NASA CONTRACTOR REPORT

NASA CR-11

NASA CR-1179

ADAPTIVE SIMULATION USING MODE IDENTIFICATION

by Rob Roy, C. H. Schley, and A. R. Axelrod

Prepared by
RENSSELAER POLYTECHNIC INSTITUTE
Troy, N. Y.
for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • NOVEMBER 1968

NASA CR-1113

ADAPTIVE SIMULATION USING MODE IDENTIFICATION

By Rob Roy, C. H. Schley, and A. R. Axelrod

Distribution of this report is provided in the interest of information exchange. Responsibility for the contents resides in the author or organization that prepared it.

Prepared under Grant No. NGR-33-018-014 by
RENSSELAER POLYTECHNIC INSTITUTE
Troy, N.Y.

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – CFSTI price $3.00

FOREWORD

The work performed under NASA grant NGR-33-018-014 covered a wide range of subjects which are coupled by the common theme of dual control. Dual control is the problem of optimal control of a process under the condition of incomplete information. Consequently, the problems of identification, adaptation, and sensitivity of optimal control systems were investigated. The final report for this grant was divided into five separate reports. The four other reports are as follows:

Error Correcting Learning Models (N68-23599 - NASA CR-94583)

Sensitivity Design Technique (N68-19267 - NASA CR-93527)

Bending Frequency Identification (Saturn Booster)
With a Digital Coherent Memory Filter (N67-39228 - CR-89319)

Pulse Rate Adaptive Threshold Logic Units (NASA CR-1035)

Adaptive Simulation Using Modal Clustering

Formulation of Problem

The subject of this report is the formulation of an input-output process model using only the process operating record. The processes considered are those which have a finite settling time. Other than a knowledge of the process settling time and the process operating record, no other information is available. The structure of the process is not known, thus nonlinear and linear processes fall within the class of processes studied.

Since the process structure is unknown, there is no procedure for obtaining the process parameters and the process nonlinearities (if any). Consequently, an exact model of the process cannot be obtained. However, an input-output model can be obtained,[1] such that given a particular input the process output can be found. A model of this type can be searched in fast time for use in a predictive control system.[2]

In addition, it is desirable that such a model be able to adapt to process changes. Since the process parameters are not monitored, the overall cause-effect relationship of the model must change, based only on the on-line operating record of the process.

Another viewpoint, and one which will be taken in this report, is that what is actually desired is an inverse model, one which portrays the output-input relationship of the process. A model of this type can be searched in fast time to obtain the input information required to guide the process through a desired output path.

In summary, the model which is obtained protrays the output-input causal relationship of a process with finite settling time. It is assumed that the only information which is available is the process settling time and the normal process operating record. Consequently, the identification is on-line with the model adapting to process changes.

Process Identification and Pattern Recognition

The problem of modeling a finite settling time process on an input-output (or output-input) basis can be viewed as a problem in pattern recognition. If the settling time of the process is T_s, then the input $x(\tau)$ from $\tau = t - \tau_s$ to $\tau = t$ uniquely determines the output $y(t)$ at time t. The input can be viewed as a two dimensional pattern on an amplitude-time coordinate system. Alternatively, if the input is sampled at N points during the process settling time then the input can be represented by a point in Euclidean N-dimensional pattern space. The output $y(t)$ can be viewed as a point in one dimensional space, the real line. Thus, the process represents a transformation or mapping from the N-dimensional input pattern space onto the real line. For a linear process the transformation is linear, for a nonlinear process the transformation is nonlinear. Pattern recognition is the process whereby a point in pattern space is mapped onto a discrete axis of categories. For example, many handwritten 2's are mapped into a single point "2" on the category axis. Similarly, many process input patterns can be mapped into a unique $y(t)$ on the real axis. If the process output $y(t)$ is quantized into R output levels, then the mapping is onto a discrete category axis of R categories. The problem of pattern recognition and of process identification is to determine the transformation process whereby input patterns are mapped onto the category axis. Furthermore, process identification seeks to determine the inverse transformation whereby a given output point can be mapped into many input patterns.

Since the art of pattern recognition has progressed to the stage where it is a usable tool, it seems natural to apply these techniques to the problem of process identification. Pattern recognition represents a particularly useful and powerful technique in the area of control systems. The control field is just realizing the importance of this approach, and investigations are presently being undertaken in the application of pattern recognition to decision making control systems.[3,4]

Category Determination

A convenient way of handling the input past is represent it in terms of an orthonormal expansion. A particularly useful set of orthonormal functions are the cardinal functions. The use of these functions enables the input to be represented in terms of the sample values of the input $x(t)$, $x(t-T), \ldots$ $x\left[t - (n-1)T\right]$; $(n-1)T = T_s$. Thus,

$$x(t - T_s, t) = \sum_{i=1}^{n} x_i(t) \; \mathcal{Q}_i(t) \tag{1}$$

where $x_i(t) = x(t - (i-1)T)$

$$\mathcal{Q}_i(t) = 2W \frac{\sin 2\pi W \, (t - (i-1)T)}{2\pi W \, (t - (i-1)T)} \qquad W = \frac{1}{2T}$$

Consequently, the input past from $t - T_s$ to t can be viewed as a point (or vector) in an n dimensional orthogonal coordinate space. This space is called pattern space. Each coordinate in pattern space represents a different time sample of the input. This vector can be written as

$$\underline{X}(t) = \text{col}\ (x_1(t),\ x_2(t),\ldots,x_i(t),\ldots,x_n(t))$$

where

$$x_i(t) = x(t - (i-1)T)$$

$$\underline{X}(t) = \text{input pattern}$$

Since the input pattern $\underline{X}$ represents the time samples of $x(t)$ during the input past, this representation remains unchanged between time samples. Consequently, the output of the process model remains unchanged between time samples. Hence the output of the process model looks like a series of steps, changing in value only at the sampling times. The height of the step at each sample time is determined on the basis of minimum mean square distance between the output of the process and the output of the model.

Let $(Z_j/\underline{X}_j)$ = output of the model given the particular input vector $\underline{X}_j$. During the time that $\underline{X}_j$ is present Z_j is a constant.

$y(t)$ = process output

The mean square distance between the process output and the model output is given by

$$\overline{D^2} = \overline{(y(t) - Z_j/\underline{X}_j)^2} = \overline{y^2(t)} - \overline{2(y(t)\ Z_j/\underline{X}_j)} + \overline{(Z_j^{\ 2}/\underline{X}_j)} \qquad (2)$$

This distance is a minimum when

$$\frac{\partial\, \overline{D^2}}{\partial\, Z_j} = 0 = - \overline{(y(t)/\underline{X}_j)} + (Z_j/\underline{X}_j)$$

or

$$(Z_j/\underline{X}_j) = \overline{(y(t)/\underline{X}_j)} \qquad (3)$$

Thus the model output should be equal to the average value of the process output when the particular input pattern appears. Since the output of the model is limited to R categories, the correct category is chosen by finding the quantization level of the average process output.

$$R_j = \text{correct model category} = Q_R \left[\overline{y(t)/\underline{X}_j} \right] \qquad (4)$$

where

Q_R = quantization operator of R quantization levels

Mode Learning Machine

The preceding section described the procedure whereby a given input pattern is assigned to a particular category. This is only part of the problem, equivalent to observing a list of patterns and their correct categorization. It would be inconceivable to construct a model which listed all possible input patterns and their associated categories. Consequently, some form of decision surface between patterns in pattern space must be constructed. These surfaces divide pattern space into regions such that known samples in a category are enclosed within a surface or region, and all other samples are excluded from this region. Theoretically, these surfaces can be constructed if the conditional probability density functions of each category are known. Using decision theoretic methods, the correct categories are then chosen on the basis of likelihood ratios.[5] Complicating the problem of constructing these surfaces is the requirement that the machine must learn in real time.

Consequently, analytic methods of determining these surfaces are out of the question. The computations and memory capacity required are too great to consider a true likelihood computer. Analytic methods for approximating the conditional probability densities are available, however investigation has shown that the learning and computation time required preclude even these analytic approximations.[6] Therefore nonanalytic methods must be used to obtain these separating surfaces. These surfaces must be obtained in real time, using each new pattern sample to correct the shape of the surface. This requirement requires a compromise between the desired decision theoretic approach and the practical considerations of simplicity and computational speed.

Since the class of processes include both linear and nonlinear processes, certain simple techniques such as linear decision functions or multilayered linear decision functions cannot be employed. However, the class of pattern recognition machines known as "modal machines" can be employed. It will now be shown that machines in this class approach the desired likelihood computers, yet maintain the advantages of simplicity and computational ease.

Consider the case of two category pattern recognition. A minimum distance classifier would assign a category to the input pattern based upon its proximity to the nearest known member of a class. The locus of points equidistant from the nearest members of the two classes forms the decision boundary. This is shown in Figure 1. The decision rule is then

$$\underline{X} \in R_1 \quad \text{if} \quad \min \left| \underline{X} - \underline{X}_{1m} \right| < \min \left| \underline{X} - \underline{X}_{2K} \right| \tag{5}$$

$\underline{X}_{1m} = m^{th}$ sample of class R_1

$\underline{X}_{2K} = K^{th}$ sample of class R_2

$\underline{X}$ = input pattern

Note that this procedure is valid no matter what the shapes of the regions which contain samples. Thus, classification is possible when samples of a given class occupy several disjointed regions, as shown in Figure 1. If the class of decision functions were limited to hyperplane boundaries, perfect class separation would not be possible.

This classification procedure has certain shortcomings. The most serious of these shortcomings is the sensitivity to stray class samples. A stray sample (due possibly to a noisy measurement) falling within an incorrect class boundary can cause numerous classification errors. Consequently, it is advantageous to modify this procedure such that, instead of looking for the sample nearest the input pattern point, a local majority rule is used. A local majority decision procedure first examines all samples within a radius r of the input pattern and counts the number of samples of each category that lie within this radius. The correct category is chosen to be that category which has the maximum number of samples within this radius. Essentially, this procedure is measuring local conditional probability, and is sometimes referred to as the Fix and Hodges procedure.[7] A modification of this procedure,[5] which weights the distance from the pattern to the stored samples, is given by

$$g_i(\underline{X}) = \text{discriminant function of } i^{th} \text{ category.} \tag{6}$$

$$= \sum_{m=1}^{N_i} \left[\frac{1}{1 + \left(\frac{|\underline{X} - \underline{X}_{im}|}{r} \right)^k} \right]$$

where

N_i = number of samples of category i within a radius r of pattern $\underline{X}$

$\underline{X}_{im}$ = m^{th} sample of category i

k = exponent which determines how pattern mismatches are weighted. Effectively k determines the slope of a filter about the point $\underline{X}$

The category to which pattern $\underline{X}$ is classified is that category with the largest discriminant function. Since this procedure is a form of a weighted likelihood computer, decisions rendered by such a technique approach Bayes decisions.

Unfortunately, this technique suffers from the disadvantage that it requires the storage of the entire learning or sample set. A reduction in the storage requirements can be accomplished if, instead of the entire sample set being stored, certain "representative" samples were stored. These representative samples can be obtained by clustering the points in the sample set. The center of each cluster is the best approximate for that population of sample points. If a new pattern point is received which is within a certain distance of the representative point, then it is assigned to that cluster and the representative point modified. Figure 2 shows how a given sample set is approximated by a union of circles. The region is then approximated by the center of the circles, with a weighting factor which indicates how many samples were contained in the cluster. These clusters can be termed "modes" of a given category. The use of these modes eases the storage requirements and simplifies the computational problems, since only the modes are stored and compared with the input pattern. A machine which clusters the input patterns and uses only modal information to effect decision making is known as a "modal machine".

Decision Making

The basic decision to be made is "given the input pattern $\underline{X}$, which category R_i is most likely". This can be transformed into an examination of the conditional probability distribution $P(R_i/\underline{X})$. Using the modal approximation of the category regions,

$$P(R_i/\underline{X}) = \frac{P(\underline{X}/R_i)\,P(R_i)}{P(\underline{X})} = \frac{\sum_{j=1}^{M_i} P(R_{ij})\,P_{R_{ij}}(\underline{X})}{P(\underline{X})} \qquad (7)$$

where

M_i = number of modes of category i

R_{ij} = j^{th} mode of category i

$P_{R_{ij}}(\underline{X})$ = conditional probability density of mode R_{ij}

Since $P(\underline{X})$ is common for all categories, category decisions can be made by comparing

$$P'(R_i/\underline{X}) = \sum_{j=1}^{M_i} N_{ij}\,P_{R_{ij}}(\underline{X}) \qquad (8)$$

where

N_{ij} = number of samples in mode R_{ij}

The probability density function $P_{R_{ij}}(\underline{X})$ is not known, but it can be approximated by a knowledge of the clustering procedure. The circles shown in Figure 2 are assumed to be equiprobable contours of Gaussian processes which have equal variances in all dimensions and uncorrelated variables. The probability density $P_{R_{ij}}(\underline{X})$ is then given by

$$P_{R_{ij}}(\underline{X}) = \frac{1}{(\sqrt{2\pi}\,\sigma)^N}\ \exp\left(-\frac{(\underline{X} - \underline{P}_{ij})^T (\underline{X} - \underline{P}_{ij})}{2\,\sigma^2}\right. \qquad (9)$$

where

N = number of dimensions

σ^2 = variance of the mode

$\underline{P}_{ij}$ = mean of the mode, referred to as the <u>prototype point</u> of the mode

Notice that if the input pattern $\underline{X}$ is close to a prototype point, all but a few terms vanish in Eq. (9). Thus, Eq. (8) is a measure of the number of samples of category i which are near $\underline{X}$, weighted according to their distances from $\underline{X}$. Hence, this type of decision approximates likelihood ratio decisions.

Alternatively, for the assumptions* made in Eq. (9), a simpler type of decision making[6] can be used. This decision procedure is a minimum distance classification based upon distances to the prototype points. The basis for this decision making is to find that category for which

$$(\underline{X} - \underline{P}_{ij})^T (\underline{X} - \underline{P}_{ij})$$

is a minimum.

Expanding this distance measure

$$(\underline{X} - \underline{P}_{ij})^T (\underline{X} - \underline{P}_{ij}) = \underline{X} \cdot \underline{X} - 2\, \underline{P}_{ij} \cdot \underline{X} + \underline{P}_{ij} \cdot \underline{P}_{ij}$$

Equivalently, the minimum distance classification within a given category can be performed by comparing the sub-discriminant functions

$$g_{ij}(\underline{X}) = \underline{X} \cdot \underline{P}_{ij} - \frac{1}{2} \underline{P}_{ij} \cdot \underline{P}_{ij} \qquad j = 1, 2, \ldots, M_i \qquad (10)$$

M_i = number of modes in category i

*The additional assumption, which may be implied from the other assumptions, is that the population of each mode is equal. This assumption is not required for this procedure, but illustrates that decisions made by both methods can be made identical.

The largest sub-discriminant function corresponds to the mode (within category i) which is closest to pattern $\underline{X}$. The correct category is obtained by comparing the discriminant functions

$$g_i(\underline{X}) = \max_{j=1,2,\ldots M_i} \left\{ \underline{X} \cdot \underline{P}_{ij} - \frac{1}{2} \underline{P}_{ij} \cdot \underline{P}_{ij} \right\} \quad i = 1, 2, \ldots, R \qquad (11)$$

and selecting the largest. The largest of the R discriminant functions is associated with the correct category. A simplified block diagram of this type of learning machine model is shown in Figure 3.

Adaptive Modal Construction

The problem of on-line construction of the modes (prototypes) of each category is handled in the following manner. The first pattern received belonging to a particular category is assigned as the first mode of that category and given a weight of one. The second pattern received belonging to that category is tested to see if it lies within a given radius (distance) of the first pattern. If it does, then it is clustered with that pattern by averaging, and a weight of two is assigned to the averaged pattern. If the second pattern falls outside of the given radius, then it is assigned as the second mode of that category. Successive patterns are treated in the same manner, clustering the patterns within a given radius so that the prototype pattern (mode) represents the center of gravity of the clustered patterns. The distance between each mode and the new pattern must be found to determine whether the new pattern should be clustered, and with which mode it is to be clustered.

Given a memory which can store M modes, the assignment of the maximum number of modes for each category raises an interesting point. The simplest solution would be to preassign to each category $\frac{M}{R}$ modes, where R is the number of categories. However, this assumption of uniform modal distribution is generally not valid, although it may produce acceptable error rates. Ideally, the number of modes assigned to each category should be chosen based upon the distribution of patterns in each category. Since the modal distribution is unknown, the assignment of modes must be made on an adaptive basis. One procedure would be to assign the incoming patterns to their respective categories, tagging each mode with the correct category. This procedure would continue until the allowed memory space was filled. In addition, a minimum number of modes could be assigned to each category.

The principal problem arises when the memory is filled. How should the M + 1st pattern be handled if it is not within the clustering radius? Conceivably, if this pattern is averaged with the nearest mode in the same category. Alternatively, the resultant mode might move the original mode away from the desired surface and closer to the inside of the category region. Therefore, once the memory is filled, averaging modes to allow for more memory space should be approached with considerable care. The new mode must be tested to see if it lies within the given category, and if it lies outside the clustering radius of any other mode. This testing of averaged modes requires a good deal of computer time, and the additional refinement of the separating surfaces must be weighed against the cost of the refinement. If the initial M modes fairly adequately represent the category surfaces, restricting the learning procedure to averaging only within the mode clustering radius may be the best possible procedure. Additional input patterns which lie outside of any clustering radius are considered as stray patterns and are discarded.

There are several other ways in which clustering can be achieved,[5,7] however this study used the simplest techniques, to evaluate the orders of magnitude achievable by use of a learning machine. A flow diagram of the simple uniform clustering procedure is shown in Figure 4.

The similarity between this method and the method of Potential Functions[8,9] should be noted. The method of Potential Functions can be viewed as either a generalization of a Φ machine[7] or as a generalization of a modal machine. In fact, Aizerman[8,9] gives two learning algorithms, one for each viewpoint or learning machine structure. The convergence proofs for these algorithms are also given. Consequently, the method used in this report can be considered to be a special case of the method of Potential Functions.

Since the purpose of this report is to illustrate how pattern recognition techniques can be used for process identification, a convergence proof for this particular form of learning machine is not included. Convergence is assured by the convergence of the general case. A similar study of process identification using the general form of the Potential Function forms the second half of this report.

Test Results

The modal learning technique was applied to a wide range of systems, both linear and nonlinear. The more important results and implications will be reported here, with the plant of Figure 5 used as an example. The input to the plant was exponentially correlated (ρ = .707), zero mean, Gaussian noise. The input pattern to the model consisted of ten sample points, taken over a five second settling time, unless otherwise noted. Results are listed in Tables I-IV. A typical output is shown in Fig. 6.

I. Uniform Prototype Model

The uniform prototype model sets aside an equal number of prototypes for each quantization category. The results of using this type of model indicated that:

1. For $\sigma = 0.5$ and 40 quantization levels, there is very little difference between using five and ten prototypes per quantization category. There is also very little difference between center of gravity clustering and non-center of gravity clustering. The rms error over 100 settling times varied between .030 and .039. The lower error was obtained for 10 prototypes, center of gravity clustering and a tolerance (radius about each prototype) of 2.0. Since a quantization interval was .025, this can be considered to be good identification, the model output being slightly higher than one quantization interval.
2. The identification time, although not shown in the tables, appeared to be within 10 settling times of the system. This is not an adequate measure since identification time in such a system is a function of the degree to which the input probes the allowed pattern space. Consequently the identification is a function of the input statistics. For the zero mean Gaussian noise the exponential

correlation (ρ = .707) provides what is seemingly rapid identification. This apparently good identification is caused by the fact that the last input pattern is closely related to the present input pattern. Consequently the present output will be closely related to the previous correctly tagged input. If the input were purely random, jumping throughout pattern space, this would not be the case. The use of exponentially correlated noise is justified on the basis of a closer match with actual signal conditions.

3. When the input variance is increased to 0.75 the mean square error increased markedly. Increasing the number of categories and increasing the number of taps decreased the error by about one-half. The principle fault in these tests was that the output range was not correspondingly increased with the input variance. Clearly, changing the input variance will change the output dynamic range. This was not accounted for in these tests. However this does point out the shortcoming of this type of identification. For a nonlinear system the entire pattern space must be probed, which requires a great deal of computer time. Equivalently, an increase in the input variance corresponds to a decrease in the number of categories or fineness of identification.

4. Use of a non-zero tolerance (radius of clustering) factor reduces the error. There is an optimum clustering radius beyond which the error increases. However this is not a sharply defined radius, as the error increases slowly after the optimum radius is exceeded.

II. Nonuniform Mode Distribution

This distribution is determined by the manner in which the input patterns are sequentially allocated to each category. A minimum of two prototypes per category are allowed, otherwise the number of prototypes per category are determined by the actual distribution.

1. For $\sigma = 0.50$ there appears to be a threshold effect in the number of categories selected. Above 50 categories (below $q = .02$) there is not any noticeable decrease in the MSE. In one case an exponential "forgetting" function was applied to the input, and in another case the input was quantized. There did not appear to be any noticeable changes for these cases. Identification was within 2 quantization levels.
2. For $\sigma = 0.75$ there was a decrease in MSE for an increase in the number of prototypes, but not a significant decrease. Consequently most of the tests were made with the lower number of prototypes. Interesting the output range was varied from 1 to 10 without any significant difference in the MSE.
3. Again, there is a combination of tolerance, range, and number of prototypes which yields a minimum MSE. There is also a unique value of system settling time which gives a minimum MSE. The order of magnitude of the MSE is largely a function of the input variance. The input variance describes the space over which the input patterns occur, the larger the variance the greater the nonlinear range which must be described by a fixed number of prototypes.
4. It is indeed interesting to note that the nonuniform distribution was not any more successful than the uniform distribution. The reason for this is described in the Conclusions Section.

Conclusions

The purpose of this study was to determine the usefulness of mode seeking pattern recognition techniques in obtaining adaptive models of nonlinear processes. This study has shown that such a simulation can be successful. However the specific form of this type of solution maps the entire input space by a set of clustered input points, each cluster belonging to an output quantization level. As such, not only the boundaries between output levels are determined but also the entire space within the boundaries. This is an inefficient use of the memory allocation, as is borne out by the increase in MSE as the input variance is increased. A more appropriate mode seeking technique would be one which cleared the inside of a region, leaving only those modes which are required to determine the boundaries between output levels. This is a much more difficult procedure than simply storing and clustering the patterns belonging to a particular class. However it should be possible to construct such boundaries by use of a polynomial decision surface which are derivable from the stored cluster patterns.

References

1. Roy, R. and Nutting, B. W., "A Hybrid Computer for Adaptive Nonlinear Process Identification," Fall Joint Computer Conference, October 1964, pp. 527-537.

2. Kaufman, H. and DeRusso, P. M., "An Adaptive Predictive Control System for Random Signals," IEEE Transactions on Automatic Control, October 1964, pp. 540-545.

3. Knoll, A. L., "Experiments with a Pattern Classifier on an Optimal Control Problem," IEEE Transactions on Automatic Control, October 1965, Vol. AC-10, No. 4, pp. 479-481.

4. Sklansky, J., "Learning Systems for Automatic Control," IEEE Transactions on Automatic Control, January 1966, Vol. AC-11, No. 1, pp. 6-19.

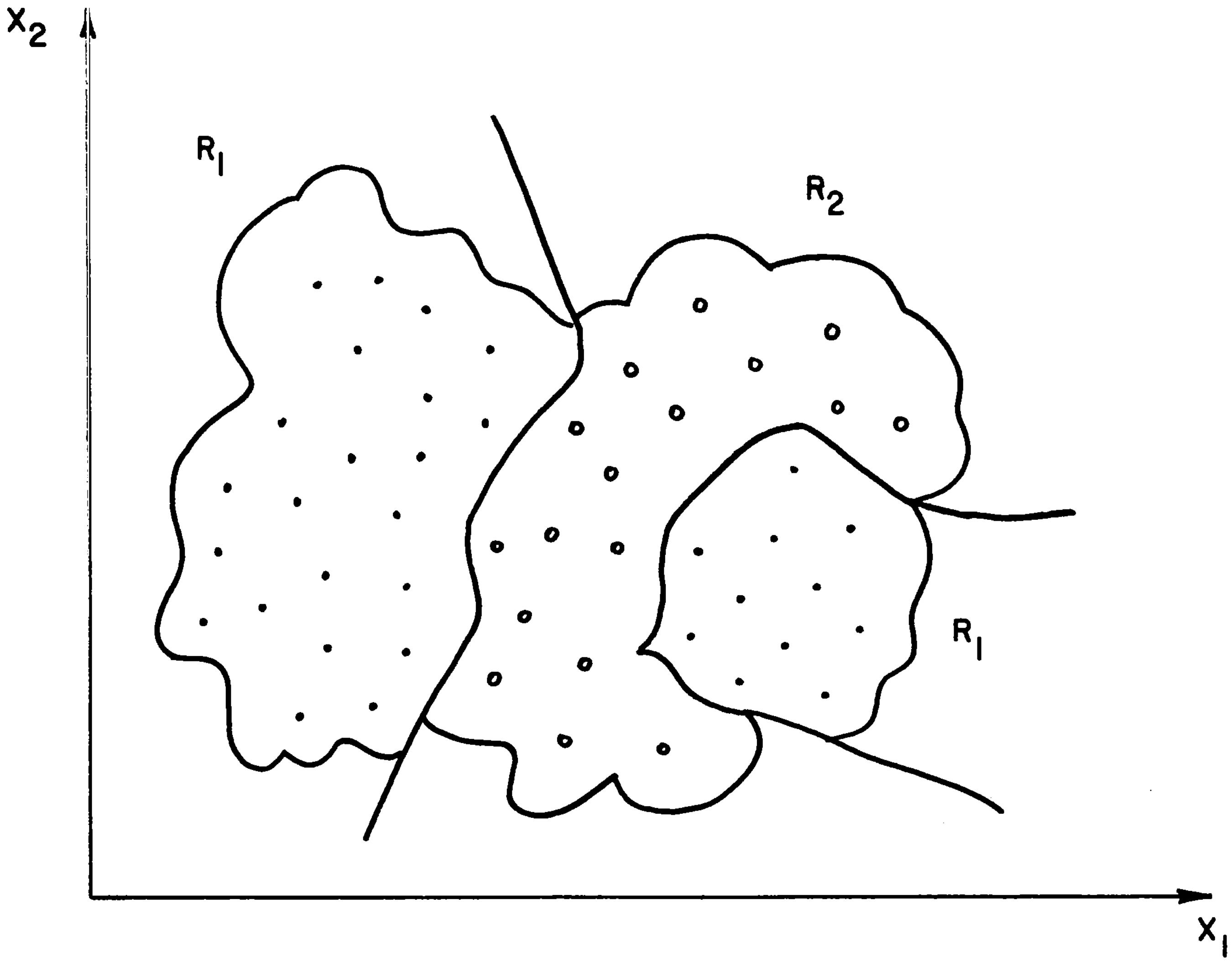
5. Sebestyen, G. S., "Decision-Making Processes in Pattern Recognition," MacMillan Co., 1962.

6. Roy, R. and Miller, R. W., "Nonlinear Process Identification Using Statistical Pattern Matrices," Fifth Joint Automatic Control Conference, June 1964, pp. 349-355.

7. Nilsson, N. J., "Learning Machines," McGraw Hill, 1965.

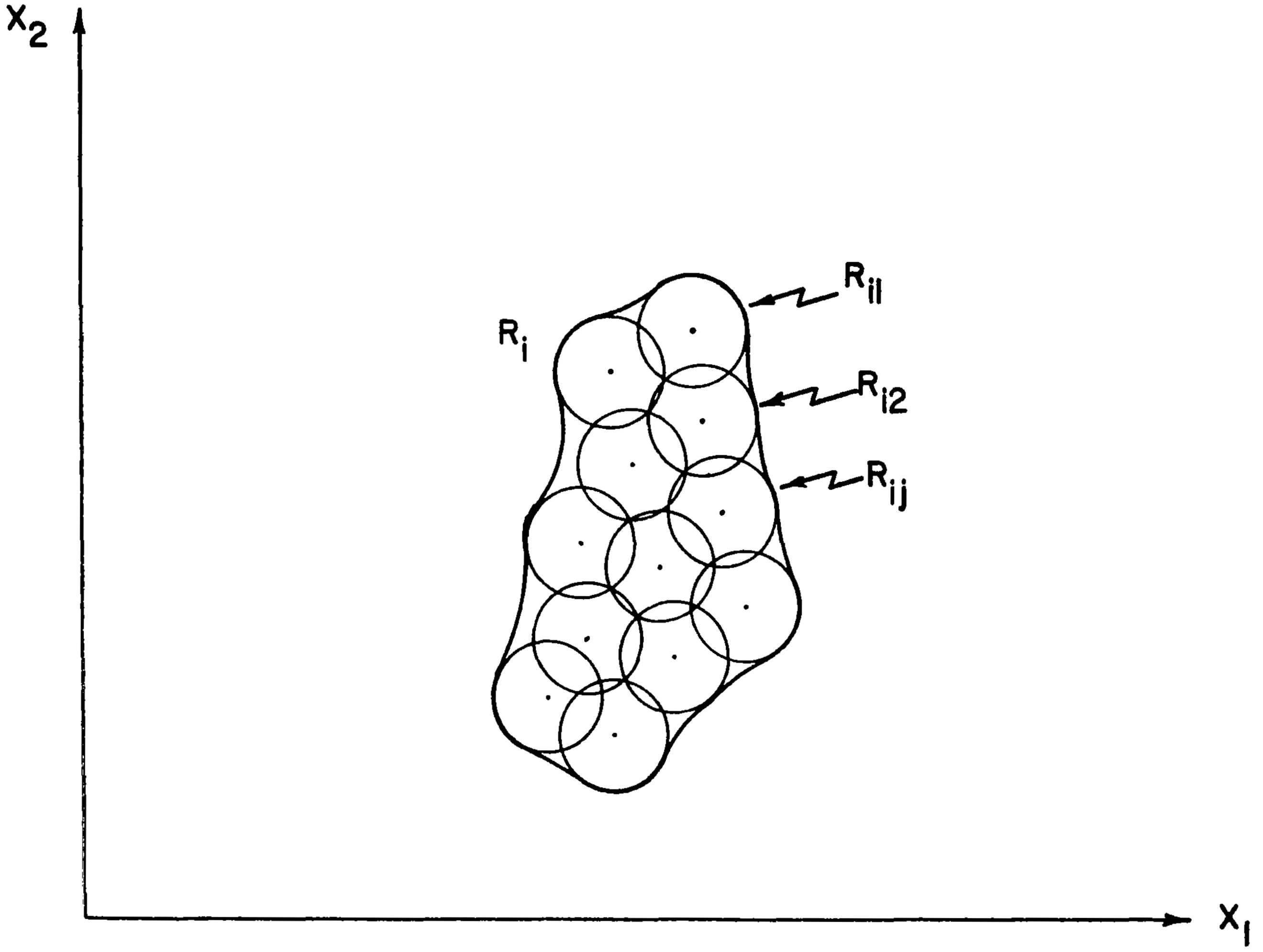
8. Aizerman, M. A. Braverman, E. M. and Rozonoer, L. I., "Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning," Automation and Remote Control, Vol. 25, No. 6, June 1964, pp. 917-936.

9. Aizerman, M. A. Braverman, E. M. and Rozonoer, L. I., "The Probability Problem of Pattern Recognition Learning and the Method of Potential Functions," Automation and Remote Control, Vol. 25, No. 9, September 1964, pp. 1307-1323.



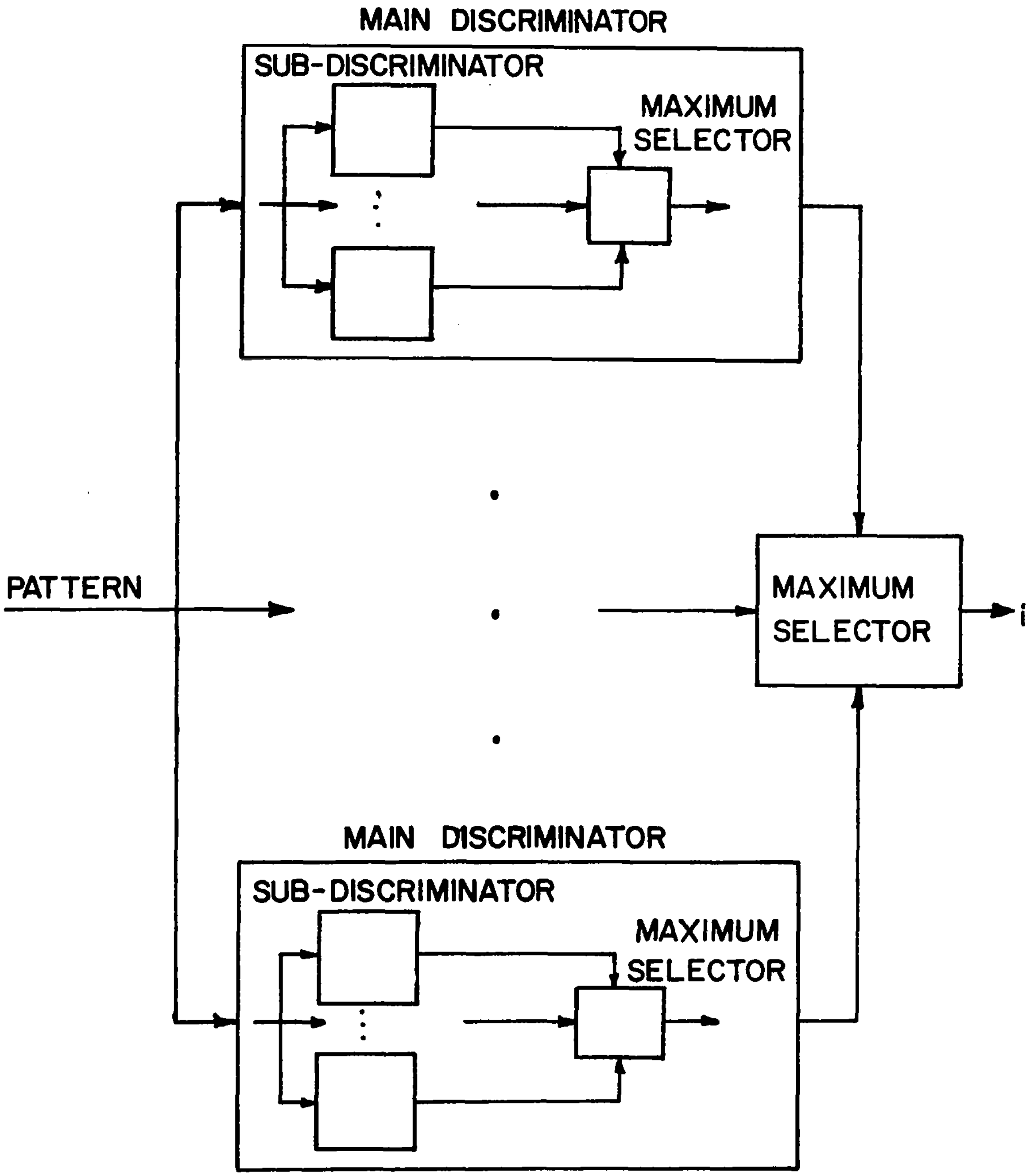
MINIMIMUM DISTANCE CLASSIFICATION

FIGURE I



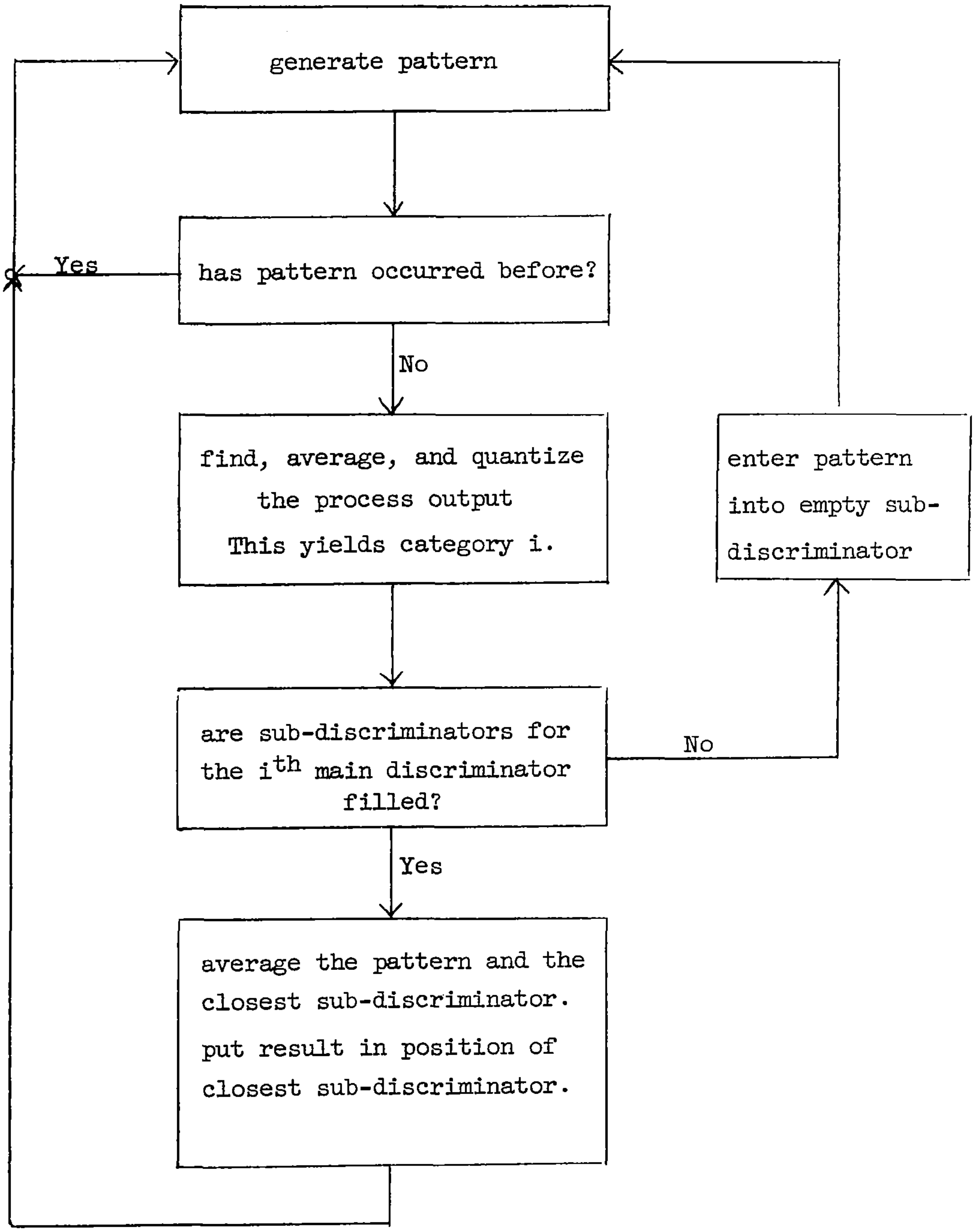
CLUSTERING AND MODE DETERMINATION

FIGURE 2



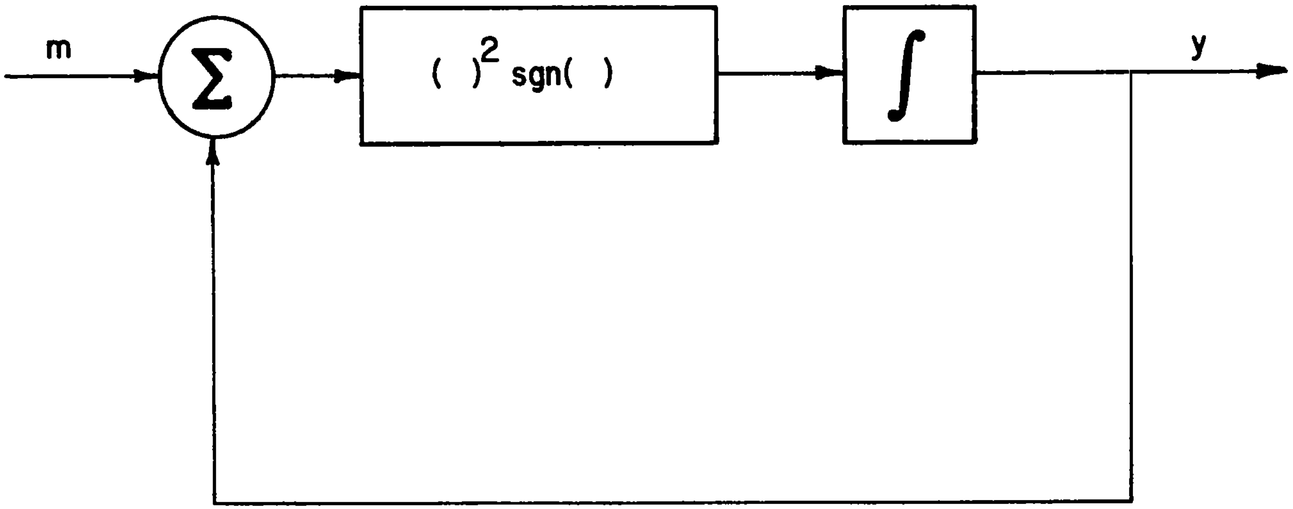
MODAL LEARNING MACHINE

FIGURE 3


generate pattern
has pattern occurred before?
Yes
No
find, average, and quantize the process output This yields category i.
enter pattern into empty sub-discriminator
are sub-discriminators for the i^{th} main discriminator filled?
No
Yes
average the pattern and the closest sub-discriminator. put result in position of closest sub-discriminator.

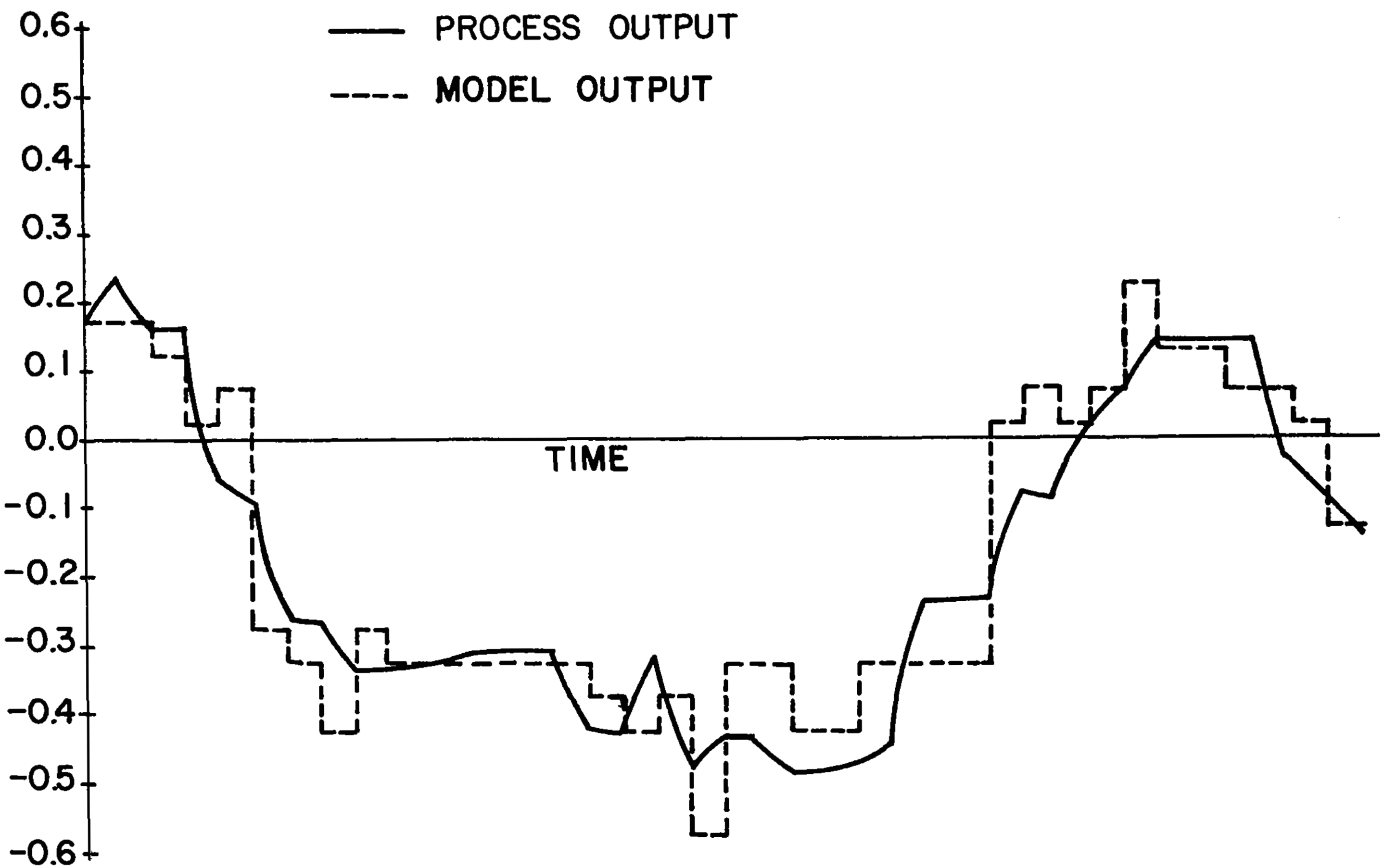
FLOW CHART FOR LEARNING PROCEDURE

FIGURE 4



NONLINEAR PROCESS

FIGURE 5



PROCESS OUTPUT AND MODEL OUTPUT

FIGURE 6

Uniform Mode Distribution

Table I

40 categories, Range = 1.0, q = .025

$\sigma = 0.5$

10 Time Points

Settling Time = 5 seconds

PROT/CAT	CLUST	TOLERANCE	MSE
5	Non C.G	0	.039
10	Non C.G	0	.034
10	C.G	0	.037
10	C.G	2.0	.030

Table II

$\sigma = 0.75$ Range = 1.0

PROT/CAT	No. CAT	q	CLUST	ST	TAPS	TOL.	MSE
10	40	.025	Non C.G	5	10	0	.164
5	80	.0125	C.G	5	10	0	.203
10	50	.02	C.G	5	10	0	.103
10	50	.02	Non C.G	5	10	0	.148
10	100	.01	C.G	5	10	0	.106
10	100	.01	C.G	5	20	0	.090
10	100	.01	C.G	5	20	2.0	.074
10	100	.01	C.G	5	20	10	.125
10	100	.01	C.G	10	20	0	.142
10	100	.01	C.G	10	10	0	.163

Nonuniform Mode Distribution

Center of Gravity Clustering

Ten equally spaced input time points

Table III

σ = 0.5 No. of Prototypes = 500 Settling Time = 5 sec.

No. of Categories	Output Range	q	Tolerance	MSE	Comments
25	10	0.4	1.0	.067	
50	1.0	.02	1.0	.041	
50	1.0	.02	0.66	.040	Forgetting
75	1.0	.013	0.66	.042	
75	1.0	.013	0.66	.038	Input quantized
100	1.0	.01	1.0	.041	

Table IV

= 0.75

No. PROT	No. CAT	Range	q	Settling Time	Tolerance	MSE	Comments
500	100	10	0.1	5	.01	.189	
1000	100	10	0.1	5	.01	.162	Increased No. Prot
500	100	1	.01	5	1.0	.209	
1000	100	1	.01	5	1.0	.202	Decreased Range
500	100	10	0.1	5	1.0	.141	
500	100	10	0.1	5	5.0	.274	Increased Tol.
500	100	1	.01	7	1.0	.241	
500	100	1	.01	10	1.0	.304	Increased Settling Time
500	100	1	.01	3	1.0	.113	Decreased Settling Time
500	100	1	.01	4	1.0	.150	
500	25	3	0.120	5	1.0	.139	
500	25	3	0.120	5	1.0	.130	Forgetting
500	25	3	0.120	5	1.0	.146	Quantized Input

ADAPTIVE SIMULATION BY THE METHOD OF POTENTIAL FUNCTIONS

Introduction

The Method of Potential Functions was first announced by Aizerman et al in 1964, and elaborated on by these same investigators in two subsequent papers, (1, 2, 3). Several Russian investigators have applied the method to some optical character recognition problems (4, 5, 8, 12), but no work seems to have been done on applying the method to the problem of adaptive simulations. In the discussion to follow, and in the investigations reported in a following section only the basic algorithm is considered.

The chief value of the Method of Potential Functions is two-fold.

1. It represents per se an algorithm which may be of considerable value in pattern recognition problems;
2. It demonstrates an analogy between modal learning techniques and Φ-machine techniques which generalize many of the results for learning in linear machines to modal machines.

The Method of Potential Functions will now be discussed. First, a general description of the method will be presented, followed by the method of realizing the algorithm. Next Aizerman's proof of the convergence of the algorithm for the two category case will be extended to R categories.

Consider an M-dimensional pattern space X. Without loss of generality, this space may be assumed to be Euclidian; and hence, to possess a metric, an inner product, and, in general, all properties associated with a Euclidian space. In particular, if $\bar{x} \in X$ is a vector in X, then one may define an orthonormal set of scalar functions $[\phi i(\bar{x}),\ i=1,2,3,---]$.

Now suppose X is partitioned into two disjoint category subsets, X_A and X_B. Then the discriminant function $\Psi(\bar{x})$ may be represented by a general

$$\Psi(x) = \sum_{i=1}^{\infty} c_i \phi_i (\bar{x}). \qquad (1)$$

The principle assumption of the method of potential functions is that this infinite sum may be adaquately represented by a finite sum.

$$\Psi(x) = \cdot \sum_{i=1}^{N} c_i \Phi_i \ (\bar{x}) \qquad (2)$$

Some intuitive justification of the principle assumption is required. We note that it often happens that in the functions of physics, a truncated Fourier series may approximate closely the original function, perhaps with some "ripple". In general, the separating surface is not necessarily unique; hence, we may consider the Truncated Fourier Series to be some "rippled" approximation to some "optimum" separating surface.

The work of Cooper (6) and Sebestyan (10) are of particular interest. Cooper shows that quadric and linear surfaces are already optimum for a wide class of probability distributions. For example, Cooper shows that the hyperplane is optimal for "two unimodal distributions differing only in location and having probability density functions which are ellipsoidally symmetric and monotonically decreasing away from the mean ";while the hypersphere is optimal when the two distributions are "spherically symetric normal with different variances". The implications here are not that these statistics may be known a priori, for if they were, parametric procedures would be far more appropriate. Rather, the point is that one might be justified in taking a small number of terms in any real physical problem. This is further supported by Sebestyan's results, cited above, page 68ff.

Consider a mapping of M-dimensional space X in some N-dimensional space Z defined as follows: let $\bar{x} = (x_1, -----, x_m)$ be a point in X. Then the image $\bar{z}$ & Z of x is given by:

$$z_i = \Phi_i\ (x),\quad i=1,----,N \qquad (4)$$

and $\bar{z} = (z_1, ----, z_N)$. Hence the separating surface given by (1) may be seen to map into a hyperplane in Z.

$$\Psi\ (\bar{z}) = \sum_{i=1}^{N} c_i z_i \qquad (5)$$

Z is termed the Linearization Space of x.

Hence, for N such that (2) holds, (2) and (4) map an arbitrary separating surface $\Psi\ (\bar{x})$ in pattern space into a linear separating surface in Z. Let $\bar{\nu} = (\nu_1, ----, \nu N)$ be the parameters of the separating hyperplane in Z. Then if $\bar{z}$ & Z is the image of $\bar{x}$ & X, the dichotomy

$$\Psi\ (\bar{x}) = \sum_{i=1}^{N} c_i\ \Phi_i\ (\bar{x}) \begin{cases} >0, & \bar{x} \in X_A \\ <0, & \bar{x} \in X_B \end{cases} \qquad (6)$$

is expressed by a correlation

$$\Psi (z) = (\bar{z}, \bar{\gamma}) \begin{cases} >0, & \bar{z} \epsilon Z_A \\ <0, & \bar{z} \epsilon Z_B \end{cases} \qquad (7)$$

The Potential Function

Let $\bar{x}$, $\bar{y}$ ϵX and let $\bar{u}$, $\bar{v}$ ϵ Z be the images of $\bar{x}$ and $\bar{y}$ under the mapping (4). Define the potential function K $(\bar{x},\bar{y})$.

$$K (\bar{x},\bar{y}) = \sum_{i=1}^{N} c_i^{\,2} \phi_i (\bar{x}) \phi_i (\bar{y}) \qquad (8)$$

It is clear that the potential function is the image in X of the correlation $(\bar{u}, \bar{v})$ in Z. The basic idea of the method of potential functions is that the separating hyperplane $\bar{\gamma}$ in linearization space can be approximated in terms of a potential function in pattern space. Prototype points are learned by the machine, and these prototype points serve as the representations in pattern space of the parameters γ_1,---- γ_N.

Algorithm

First Method

The potential function K $(\bar{x}, \bar{y})$ is chosen. For example, two likely functions are of the form $A/[1 + B \,||\bar{x} - \bar{y}||^2]$ and A exp.$(-B||\bar{x}-\bar{y}||^2)$ where $||\ ||$ indicates the norm defined on X-space. As a practical matter, the norm may be taken to be Euclidean distance.

The algorithm (first method) is defined by induction, as follows: the first point $\bar{x}$ appears and the potential function is defined as

$$\Psi_1 (\bar{x}) = \begin{cases} K(\bar{x},x^1), & x \epsilon X_1 \\ -K(x,x^1), & x \epsilon X_2 \end{cases} \qquad (9)$$

Now, assume $\Psi_r(\bar{x})$ is defined. Let the point $\bar{X}^{r+1}$ appear. Then 4 cases exist:

a) $\bar{X}^{r+1} \varepsilon X_1, \Psi_r(\bar{x}^{r+1}) > 0$

b) $\bar{X}^{r+1} \varepsilon X_2, \Psi_r(\bar{x}^{r+1}) < 0$

c) $\bar{X}^{r+1} \varepsilon X_1 \ \Psi_r(x^{r+1}) < 0$

d) $\bar{X}^{r+1} \varepsilon X_2, \Psi_r(x^{r+1}) > 0$

Then $K_{r+1}(\bar{x})$ is defined as

$$\Psi_{r+1}(x) = \begin{cases} \Psi_r(\bar{x}), & \text{a) and b) } \underline{i}.\underline{e}.\text{, no error} \\ \Psi_r(\bar{x})+K(\bar{x},\bar{x}^{r+1}), & \text{c)} \\ \Psi_r(\bar{x})-K(\bar{x},\bar{x}^{r+1}), & \text{d)} \end{cases} \qquad (10)$$

After r steps, the potential function may be written in the form

$$\Psi_r(\bar{x}) = \sum_{x^s \varepsilon X_1}{}' K(\bar{x},\bar{x}^s) - \sum_{x^q \varepsilon X_2}{}' K(\bar{x},\bar{x}^q) \qquad (11)$$

The prime on the summation means that only those $(\bar{x}^s \varepsilon X_1)$ and $(\bar{x}^q \varepsilon X_2)$ are taken which caused the preceding potential function to be in error. As a matter of terminology, the sets $(\bar{x}^s \text{ E } X_1)$ and $(\bar{x}^q \text{ E } X_2)$ may be termed; respectively, the positive and negative <u>poles</u> associated with category X_1.

It is clear that in this algorithm these poles must be stored during the learning period. As $r \rightarrow \infty$, the function $\Psi_r(\bar{x})$ should converge to the separation function $\Psi(\bar{x})$, so that as learning progresses, fewer poles will be added to storage.

Equation (11) may be restated more compactly in terms of the adjusted training set $S_{\hat{X}}$. Let $S_{\hat{X}} = (\bar{x}_1, \bar{x}_2, \ldots)$ be the set of training pattern vectors which were misclassified by the potential function. The <u>Pole weight factor</u> associated with member of the adjusted training set is denoted by α_j and is defined as

$$\alpha_j = \left\{ \begin{array}{l} +1,\ \hat{x}_j \in X_1 \\ \\ -1,\ x_j \in X_2 \end{array} \right\},\ \hat{x}_j \in S\hat{X}$$

Hence, $\alpha_j = +1$ for <u>positive</u> poles, and $\alpha_j = +1$ for <u>negative</u> poles. Then equation (11) may be written as:

$$\Psi_r(\bar{x}) = \sum_{j=1}^{N_r} \alpha_j\, K(\bar{x}, \hat{x}_j),\ \hat{x}_j \in S_{\hat{X}} \qquad (13)$$

The first algorithm is then seen to be a modal technique, with the poles, which are seen to be the members of the adjusted training set $S_{\hat{X}}$, corresponding to prototype points. This number, N_r, cannot, in general, be computed <u>a priori</u>. This is a drawback in the method, since clearly, the amount of computer memory required to carry out the algorithm is directly proportional to N_r. In simulation experiments carried out employing this technique, methods arbitrarily constraining N_r were attempted, with generally satisfactory results.

<u>Extension of the Algorithm to the R-Category Case.</u>

The Potential Function Identifier is proposed as an adaptive model for non-linear systems. The patterns to be classified are sets of time samples of the input, while the categories are defined by quantizing the output signal into R levels. Several schemes for extending the algorithm are now suggested.

Absolute learning scheme.

R potential functions are defined. During learning, when x^{r+1} ϵ (level j) appears, each function $\Psi_i^{r+1}(x^{r-1})$ is corrected so that

$$\Psi_i^{r+1}(x^{r+1}) \left\{ \begin{array}{l} < 0, \ i + j \\ \\ > 0, \ i = j \end{array} \right.$$

During identification, if convergence were complete, for each·$\bar{x}$ that appears, one and only one of the R potential functions would be positive and all of the others would be negative. The positive function would be selected as that corresponding to the desired quantization level. As a matter of practice, such a condition could not in general be expected. In the case where several potential functions were positive, the most reasonable choice might be the maximum.

Maximum Learning Scheme.

In this technique, it is not required that the appropriate potential function be positive while all others be negative; only that the appropriate potential function be greater than any other. Hence, during learning when $x^{r+1}\epsilon \left(\text{level } j\right)$ appears, the maximum potential function is calculated

$$\Psi_k^{r}(\bar{x}^{r+1}) = \underset{i=1,R}{\text{Max}} \quad (\Psi_i^{r}(\bar{x}_{r+1}))$$

Then if $k=j$, (i.e., no error), no correction is made, while if $k \neq j$, a positive correction is made for Ψ_j and a negative correction is made for Ψ_k. All other potential functions are left uncorrected. In terms of the First Algorithm, the assumption is that this will result in fewer poles being stored. During identification, when $\bar{x}$ appears, the maximum $\Psi_i(x)$ is selected as corresponding to the desired category.

One final remark may be made concerning both the absolute and maximum techniques. If, during identification, a point $\bar{x}$ appears such that none of the $\Psi_i(x)$ are positive, this situation may be considered to be an error. In this case, no decision is made, and the last estimate, for lack of any better criterion is retained.

Convergence of the Algorithm.

One of the powerful aspects of the Method of Potential Functions is that it is a highly general modal technique whose convergence properties are well understood. The proof of Aizerman et al is given for the two category case. It is necessary to extend this proof to the multi-category case. In this report, Aizerman's Theorem will be stated without proof, primarily for purposes of reference; likewise, theorems which extend the result will be stated and discussed briefly; but no rigorous proof will be given.

Aizerman's Theorem for the two category is stated as follows:

Thm. In pattern space X, let the function $\Psi(\bar{x})$, $\bar{x} \in X$ separate x into two subsets, X_A and X_B such that

$$\Psi(\bar{x}) \begin{cases} > \varepsilon , & \bar{x} \in X_A \\ < -\varepsilon, & \bar{x} \in X_B , \quad \varepsilon > 0 \end{cases}$$

and let $\Psi(\bar{x})$ be representable in the form of equation (2). Let S_x be an arbitrary infinite sequence of points in x, $(\bar{x}_1, \bar{x}_2, \text{----} \bar{x}_k, \text{----})$. Let the function $K(\bar{x}, \bar{x})$ be bounded in x.

Then, there exists some integer M , independent of the choice of S_x, such that the number of correlated errors does not exceed M.

This states that the algorithm converges in a finite number of steps. However, it is noted that this upper bound M is not a priori. calculable. The implications of this are discussed below.

Extension to the Multi-Category Case.

Let there be defined in pattern space R subsets, X_1, X_2----,X_R. Assume that these subsets are disjoint, and define the training set X to be the union of these category subsets:

$$X = \bigcup_{i=1}^{R} X_i \qquad (14)$$

Then, R distinct discriminant functions $\Psi_i(\bar{x})$, i=1,R are defined and each associated with one of the R category sets.

The conditions for separability may now be stated precisely: The set of category subsets X_1,----X_R are separable in the Absolute sense (or absolutely separable) by the function set $\Psi_1(\bar{x})$,---- $\Psi_r(\bar{x})$ if and only if

$$\Psi_j\ (\bar{x}) \left\{ \begin{matrix} > 0, \ \bar{x} \ \& \ X_j \\ \\ < 0, \ \bar{x} \ \& \ X_j \end{matrix} \right\}, \ j=1,----,R \qquad (15)$$

The set category subsets X_1,----,X_r are separable in the maximum sense (maximally separable) if and only if

$$\Psi_j\ (\bar{x}) > \Psi_i(\bar{x}),\ \bar{x}\ E\ X_j;\ i,\ j=1,\ ----\ R;\ i + j \qquad (16)$$

Now, it is clear that any set of categories which is absolutely separable is necessarily maximally separable.

Analogously, these criteria can be stated in linearization space. Let Z_1,----,Z_r be the mappings of the sets X_1,----X_r, respectively. Then, if $\bar{\gamma}_j$ is the solution weight vector corresponding to $\Psi_j\ (\bar{x})$, then absolute separability yields

$$\bar{\gamma}_j \bullet \bar{z} \left\{ \begin{matrix} > 0, \ \bar{z} \ \& \ Z_j \\ \\ < 0, \ \bar{z} \ \& \ Z_j \end{matrix} \right\} \quad , \ j = 1,----,R \qquad (17)$$

and maximal separability yields

$$\bar{\gamma}_j \cdot \bar{z} > \bar{\gamma}_i \cdot \bar{z}, \text{ all } \bar{z} \in Z_j;\ i, j = 1,---,R_j\ i \neq j \qquad (18)$$

These conditions, of course, represent linear separability in Z -space.

Then the extension of the convergence proof through the multi-category follows in a straight-forward manner. Rather than state these results formally as theorems, the results will simply be discussed. For the Absolute scheme, one recognizes that R categories are partitioned by R-1 discriminant functions. Hence, this method may be thought of as R-1 2-category separations carried out in parallel, according to equation (15). Hence, the result follows immediately.

For the Maximum scheme, the convergence result is extended by noting that in linearization space, the algorithm reduces to fixed-increment linear training procedure. The Nilsson-Kesler Theorem, (10, p. 87) may be adapted directly to extend the result. For details, see reference.

Simulation of the Systems

Three systems were employed; these were a 2-dimensional system, a first order nonlinear system, and a second order system with saturation. These are now described.

Two-Dimensional Delay System

The equation for the Two-Dimensional Delay system is

$$y(t) = x(t) - x(t-T). \qquad (19)$$

This trivial system was employed precisely for the reason that the output is completely specified by only two taps on a delay line. Hence, pattern space may be taken as two-dimensional, and the actual and computed discriminant surfaces displayed.

The output was quantized into 5 levels such that

$$\begin{aligned} Y_0 &= -5 \\ Y_1 &= -3 \\ Y_2 &= -1 \\ Y_3 &= 1 \\ Y_4 &= 3 \\ Y_5 &= 5 \end{aligned} \tag{20}$$

Let the axes of pattern space be designated by $m_1 = x(t)$, $m_2 = x(t-T)$. Then the equations for the discriminant lines are

$$m_2 = m_1 - Y_i, \ i = 1, 2, 3, 4, \tag{21}$$

Since the input x(t) is gaussian, the distribution of pattern points $Z = (m_1, m_2)$ in pattern space is given by a bivariate gaussian distribution. The distributions of m_1 and m_2 were such that m_1 and m_2 have zero mean and identical σ. Hence, the equiprobabilistic contours are sketched in the pattern plane according to

$$M_1^2 - 2\rho M_1 M_2 + M_2^2 = 0 \tag{22}$$

Figure 1 shows pattern space for the 2-dimensional delay system, with equiprobable contours for $\sigma = \frac{1}{2}$, $\sigma = 1$, and $\sigma = 2$.

First order nonlinear system

The second system employed was termed the first order nonlinear system. The block diagram is shown in Figure 2.

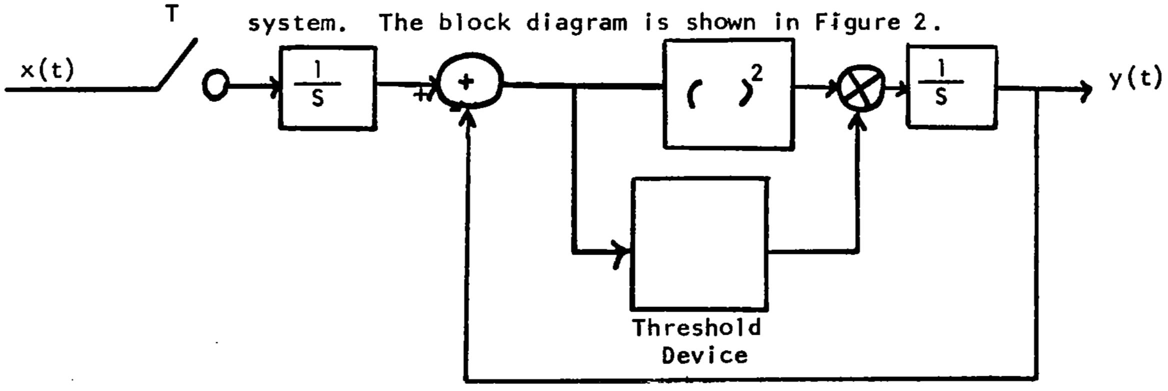
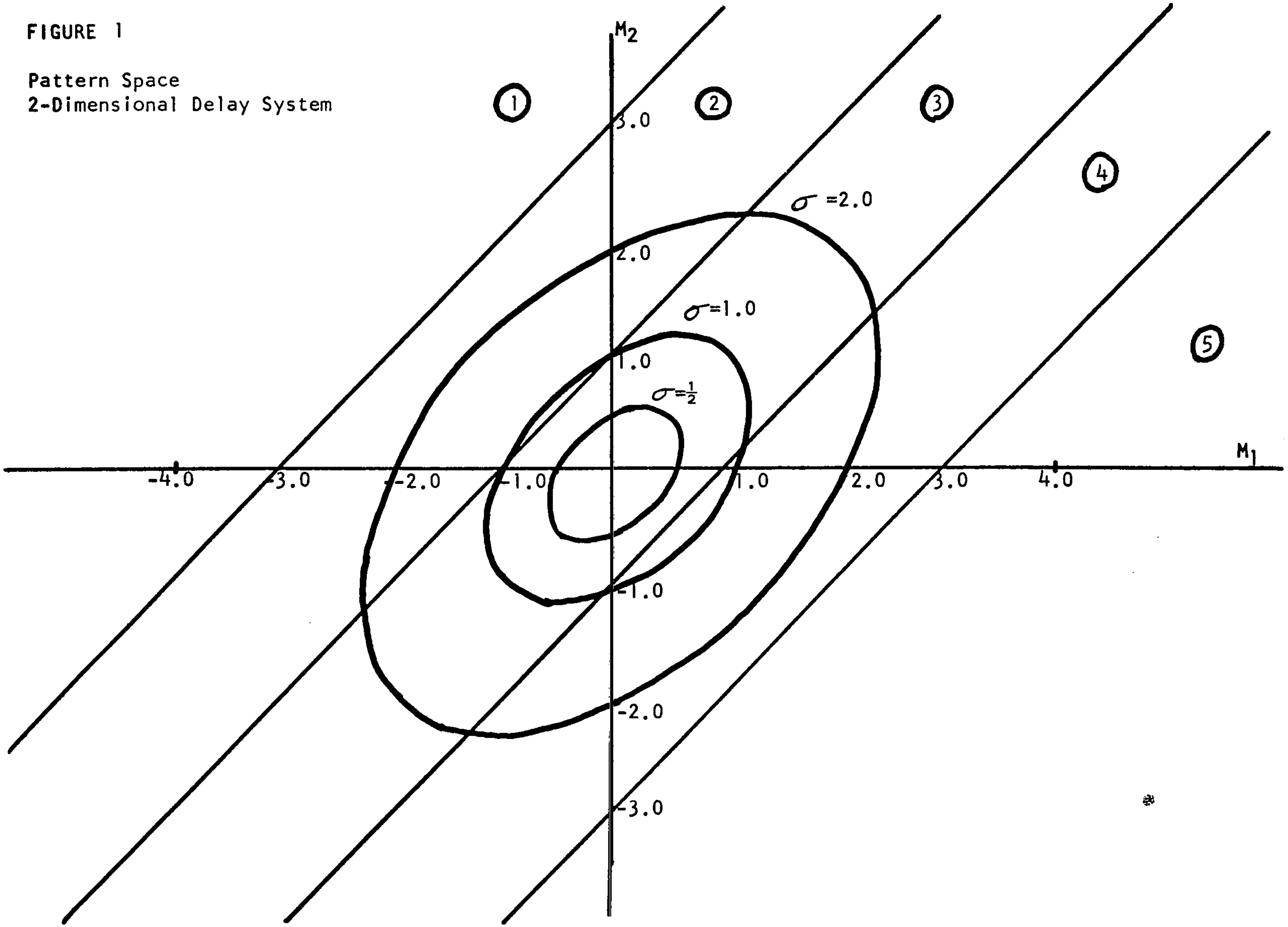


Figure 2 - First Order Nonlinear System

FIGURE 1

Pattern Space
2-Dimensional Delay System



The differential equation for the system is

$$\frac{dy}{dt} = (x - y)^2 \quad \mathrm{SGN}\ (x - y)$$

The solution to this nonlinear differential equation can be found in closed form by separation of variables to be

$$y(nT + \tau) = x_n - \frac{x_n - y\ (nT)}{1 + |\ x_n - y\ (nT)/\tau}\ ,\ 0 \leq \tau \leq T \qquad (23)$$

Furthermore, the value of $y(nT + \tau)$ averaged over the interval $0 \leq \tau \leq T$, that is, the value of $y(t)$ averaged over the interval $nT \leq t \leq (n-1)\ T$, can also be found in closed form.

$$Y_{AV}(n) = Xn - \frac{1}{T}\ \mathrm{SGN}\ (x'n - y(nT))\ \ \log\ (1 + /x'n-y$$

$$(24)$$

The step response of the FONS is given in Figure 3.

This system was employed because no numerical integration is required.

Second Order Nonlinear System

The third system employed was a second order system with a saturation nonlinearity. The block diagram is given in figure 4.

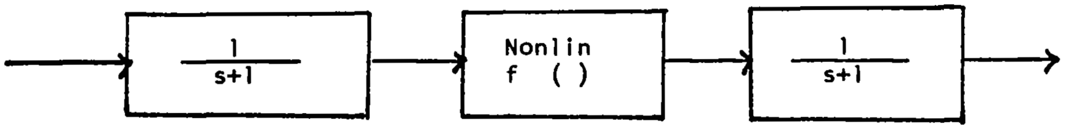


Figure 4 - Second Order Saturation System

The nonlinearity employed for studies of the potential function Algorithm was a saturation, given by:

$$f(u) = \begin{cases} \frac{1}{2},\ u \geq \frac{1}{2} \\ u,\ -\frac{1}{2} < u < \frac{1}{2} \\ -\frac{1}{2},\ u \leq -\frac{1}{2} \end{cases}$$

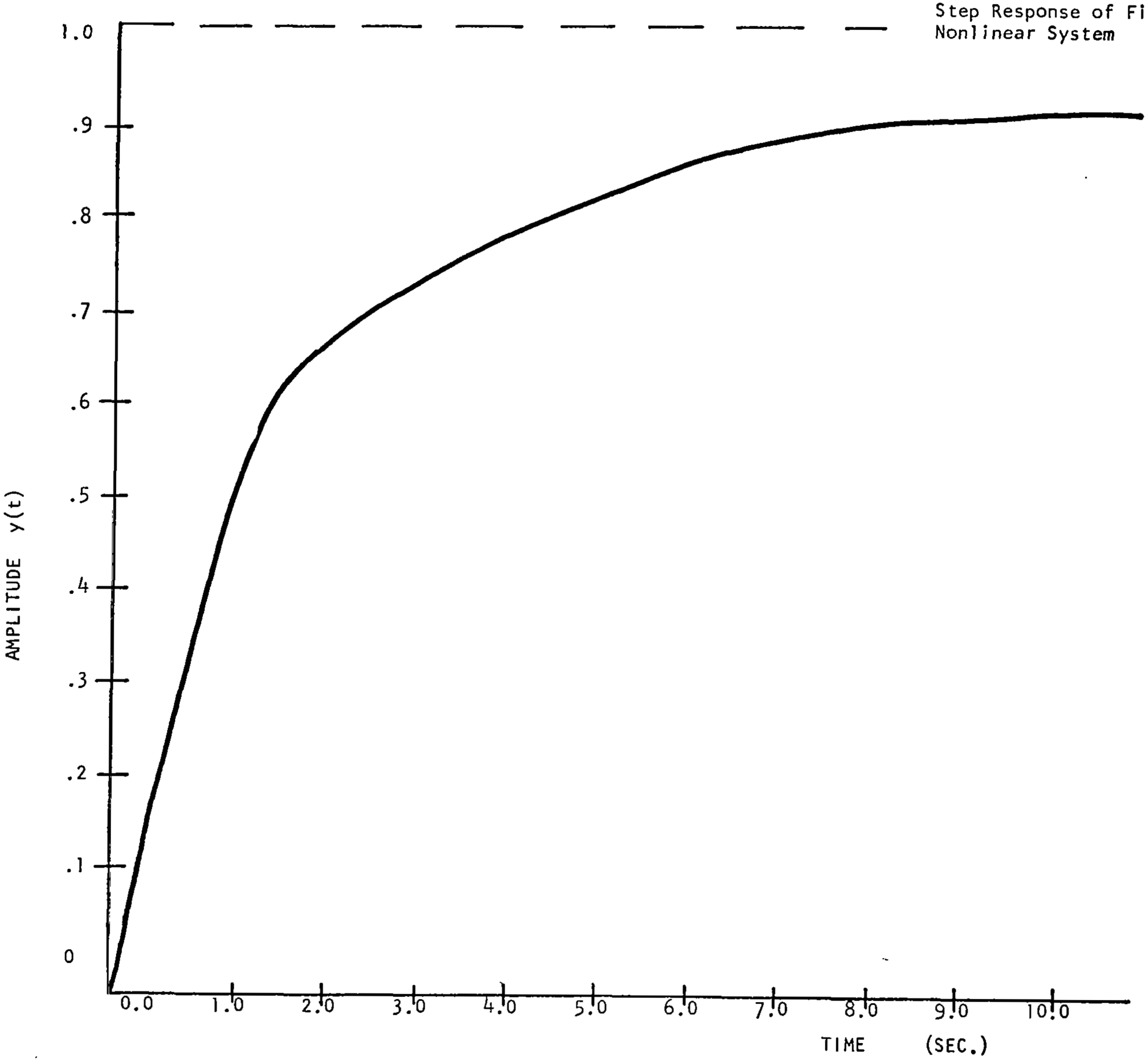


FIGURE 3
Step Response of First Order
Nonlinear System

Simulation Studies

In particular the following problems are considered:

1. Memory considerations: the memory size is determined by the number of taps on the delay line, the number of quantization levels for the output signal, and the maximum number of poles per category (that is, the maximum number of elements in the reduced category training sets). Hence, it is necessary to investigate choice of number of quantization levels and some method of limiting the maximum number of poles per level. It is, in general, assumed that the number of taps and the tap interval is fixed.

2. Choice of what is termed "the learning routine": That is, two methods of determining the reduced category training sequences were discussed. These were the Maximum Scheme and the Absolute Scheme. As has been pointed out, the reduced category training sequences are the poles of the potential function, and these two criteria, in effect, define two distinct algorithms for calculating the poles of the potential function. Accordingly, in the discussion to follow, the term "Absolute learning routine" refers to selection of poles according to equation (15), while the term "Maximum learning routine" refers to selection of poles according to (16).

3. Choice of the potential function, $K(\overline{x}, \overline{y})$: Only one potential function, referred to as the "Butterworth Potential Function" was employed. It is of the form:

$$K(\overline{X},\overline{Y}) = \frac{1}{1 + \left[\frac{\|\overline{X} - \overline{Y}\|}{R}\right]^{2 \cdot E}} \qquad (25)$$

Practical considerations

In this section the above problems, as well as several others, are discussed in further detail and the techniques of simulation are described.

Memory Considerations

In a delay line synthesizer, the required number of taps on the delay line, and the tap interval is given by:

$$N_T = 2fm\,T + 1$$

$$T \leq \frac{1}{2fm}$$

where N_T is the number of taps required, and T is the tap interval. Then the pattern vector will have N_T components,

$$\overline{X} = (x_1, ----, x_{N_T})$$

Next suppose the output y(t) were to be quantized into N_L levels. As has been discussed, each quantization level corresponds to a category subset in pattern space. Finally, suppose that it is determined that each category subset shall contain no more than N_{max} poles. Then it is clear that the computer storage required is:

$$n_p = N_T \cdot N_L \cdot N_{MAX} \tag{26}$$

In addition, an N_L X N_{MAX} matrix $\left[\alpha\right]$ is required such that $\alpha_{iK} = \pm 1$ depending upon whether the k^{th} pole in the i^{th} category is positive or negative. In a general purpose computer, the matrix $\left[\alpha\right]$ must be stored in $N_L \cdot N_{MAX}$ words; however, if a special purpose computer were built to realize the algorithm, since α_{iK} is a two-valued number, only $N_L \cdot N_{MAX}$ bits would be required.

It is noted that the number n_p is a fairly large number. For example, if $N_T = 10$, $N_L = 15$, and $N_{MAX} = 50$, then $n_p = 7{,}500$ words, which, though not unreasonable, is nevertheless large. The dependence upon these values, however, is linear, avoiding the problems encountered by several other schemes.

The Maximum Number of Poles per Category

The proof of the convergence of the Method of Potential Functions shows that there exists an upper bound, k_o, such that the reduced training sequence contains not more than k_o members, in terms of properties of the category subsets in linearization space. However, these properties are not, in general, measurable or calculable *a priori*. In short, it is not, in general, possible to determine *a priori* how many poles per category will be required; therefore, some arbitrary limitation must be placed upon computer storage by arbitrarily selecting N_{MAX}. Now, choosing N_{MAX} in such a way as to result in reasonable storage requirements does not guarantee that the algorithm will have terminated by the time N_{MAX} poles are stored, which is to say that the identification error will not necessarily be acceptably small.

To attempt to get around this problem, clustering techniques were investigated in the realization of the Algorithm. The techniques of clustering is based on the analogy of the method of potential functions to other modal techniques, described in the literature.

Description of Pole Weighting

Briefly, the clustering technique proceeds as follows: suppose the i^{th} reduced category training sequence $S^{i}_{\hat{x}}$ already contains N_{MAX} members, that is, N_{MAX} poles have already been stored. Then another pole $\overline{X}^{i}_{k}$ appears where $k = N_{MAX}+1$. Then, instead of adding the new pole to the $S^{i}_{\hat{X}}$, instead, the closest <u>odd pole of the same polarity</u> is found and replaced with a weighted average of itself and the new pole.

That is, suppose $\overline{X}^{i}_{k}$ were the new pole, where $k > N_{MAX}$, and $\alpha_{ik} = \pm 1$. The $S^{i}_{\hat{X}}$ would be searched for the pole $\hat{X}^{i}_{j}$ & $S^{i}_{\hat{X}}$ such that $\alpha_{ij} = \alpha_{ik}$ and Min $||\overline{X}^{i}_{k} - \hat{X}^{i}_{j}||$.

Then the closest old pole $\hat{X}^i_j$ is replaced by the weighted average of itself and the new pole. Symbolically:

$$\hat{X}^i_j \rightarrow \hat{X}^{i\prime}_j \qquad \text{where}$$

$$\hat{X}^{i\prime}_j = w_o \hat{X}^i_j + w_n \hat{X}^i_K$$

$$w_o + w_n = 1 \qquad (27)$$

wo and wn are two scalars which are the weights by which the new pole and the closest old pole are averaged. These weights may be selected arbitrarily which is termed uniform weighting, or they may be computed by the algorithm according to some pre-selected rule. The rule employed in the simulation was so-called center of gravity weighting rule. In this weighting procedure, the weights are selected so that if a particular pole is to be modified for the N_Wth time, according to (27), the weights Wo and Wn are set such that

$$\frac{Wo}{Wn} = N_w \qquad (28)$$

For the uniform weighting procedure, the weights wo and wn are pre-selected and remain constant. For the center of gravity (CG) procedure, a $N_L \times N_{MAX}$ matrix $[N_w\ (L,N)]$ is stored such that Nw (L,N) = 1 initially and is increased by one each time the N^{th} pole of the L^{th} category is weighted according to (27).

Clearly, the C.G. weighting procedure requires more storage and more computation than does the uniform procedure. Simulation studies suggest that this added complexity may not be justified.

Two special cases of the uniform procedure may be noted.

1. Suppose $w_o = 0$, $w_n = 1$. Thus, a new pole simply replaces the nearest old pole. This might be of advantage for nonstationary systems where the new pole is considered to be an "update" of the system.

2. Suppose $w_o = 1$, $w_n = 0$. Or rather, no averaging is done. That is, once a category is "filled", we assume we are satisfied with it, and do no further learning. For a stationary system, this results in minimum computation.

Simulation studies suggest that if N_{MAX} is large enough, weighting does not increase simulation accuracy. Hence, cases (1) or (2) may as well be selected, if appropriate.

The Potential Function

The potential function employed referred to as the "Butterworth Potential Function", was of the form given in (25). Several considerations led to the selection of a function of this form. First, intuition suggest that the potential function should be a monotone decreasing function of $\| \overline{X}-\overline{Y} \|$. This is because each positive pole may be thought of as establishing a "mode", and we desire the extent of this mode be limited. The parameters R and E define the extent of influence of the mode defined by each positive pole. R determines the "half-width" of the potential while E determines the steepness.

Furthermore, since the Butterworth Function is of great generality with respect to the "shape" of the potential function in pattern space, it was therefore the only one employed.

Realization of the Algorithm

Definition of the Discriminant Function

In the Method of Potential Functions (First Algorithm) the discriminant function in pattern space is defined by the set of positive and negative poles of the potential function. As has been shown, these poles are identically the members of the reduced category training sequences.

$S_{\hat{X}}^{i}$, i = 1,----,N_L. Let $\hat{X}_n^{\ell}$ be the nth member of $S_{\hat{X}}^{\ell}$. Then $\hat{X}_n^{\ell}$ is a positive pole, denoted by $\alpha_{\ n}$ = +1, or $\hat{X}_n^{\ell}$ is a negative pole, denoted by $\alpha_{\ell n}$ = -1.

Then the potential function discriminant is defined as follows: Let the number of levels N_L and the maximum number of poles per category N_{MAX} be chosen. Then define an N_L x N_{MAX} Vector Matrix $\bar{P}$, and an N_L x N_{MAX} scalar matrix $\bar{A}$ such that the vector P (ℓ,n) is the nth member of the ℓth reduced category training sequence, $\bar{P}$ $(\ell,n) = \hat{X}_n^{\ell}$, ℓ = 1,----, N_L; n = 1,----, N_{MAX}, and α $(\ell$, n) & A= ± 1. Note that $\bar{P}$ is in reality a three-dimensional scalar array N_T x N_L x N_{MAX}. Also define an integer array N_N (ℓ), ℓ = 1,----,N_L such that at each step of the algorithm, $N_N(\ell)$ is the number of poles already in the ℓth reduced category training sequence. Then, at each point of the learning phase of the algorithm, the N_L discriminant functions are defined by:

$$\Psi_1 \ (X) = \sum_{n=1}^{N_N(1)} \alpha \ (1,n) \ K \ (\bar{X},\bar{P}(\ell,n)), \ell=1,----,N_L \qquad (29)$$

<u>Procedure</u>

Then at the kth step of the algorithm, the pattern $\bar{X}_K$& S_X is examined according to (15) or (16), and if an error is found, then either

a. if the appropriate $N_N(\ell) < N_{MAX}$, the pattern $\bar{X}$ is added to the array $\bar{P}$, the value $N(\ell)$ is increased by one, and $\alpha(\ell,N_N(\ell))$ is set to the appropriate value; or

b. if N_N (ℓ) = N_{MAX}, the pattern $\bar{X}$ is weighted with the closest old pole of the same polarity, according to (27).

During the learning phase patterns are examined according to either the Maximum or Absolute learning scheme, and may be added to storage according to the procedure. Periodically, the algorithm branches to the identification phase, during which a standard specimen signal is identified. The identification phase differs from the learning phase in the following respects:

a. Regardless of the learning routine, identification is made with a maximum criterion. This is permissible since any set of categories which is maximally separable is also absolutely separable. Furthermore, since maximal separability is a weaker condition, fewer equivocal classifications would be made (that is, classifications where more than one, or none of the N_L discriminant functions are positive).

b. During the learning phase, the identification error (R M S) and réliability are measured. Each time the specimen signal is tested, the realization returns three quantities: the RMS error of the specimen signal, the ratio of correct identifications, and the ratio of identifications correct to within one quantization interval.

c. During the identification phase, patterns incorrectly classified are not added to storage.

It is assumed that the Specimen signal is representative of the learning signal, having the same statistics, but that the specimen signal is not identical to any segment of the learning signal.

Experimental Results

The purpose of the experiments is two-fold: first to verify that the method of potential functions is useful as a method of adaptive simulation of non-linear systems; second to provide some insight in the choice of parameters.

The parameters of interest are the number quantization levels of the output, the number of past samples of the input, the maximum number of poles per category, the pole weighting factors, and the potential factor parameters R and E. Also of interest is some comparison between the Maximum and Absolute learning routines. Finally, these factors are to be examined with respect to the statistics of the input signal, in the present case ρ and σ (in all cases, input of mean value = 0 were used)

Two-Dimensional Delay System

Choice of Learning Algorithm.

Comparison of figures 5 and 8 for the maximum routine with figures 9 and 10 for the absolute routine indicate that neither is markedly better than the other, with one reservation: That is, in figure 9, there is a region computed as category 1 that is actually category 4. This would be a more serious error than being misclassified as category 3.

Table 1 indicates that both require a comparable number of poles. The differences between the two may be 1) that the maximum routine is somwhat more accurate than the absolute one; 2) that the absolute routine may require a somewhat longer learning time, or 3) fortuitous. In any case, the difference is not marked.

Effect of Radius R.

Comparison of the cases for R = 1 and R = 4 indicates that the radius parameter is indeed an extremely important one. Note that in comparing figures 5 with 7, and figures 9 with 10, that the larger radius results in a higher pole density. This is the reverse of what would be true for other modal schemes. Note that in the region of interest, accuracy is worse for the larger R.

This is explained by the fact that the region of influence of a pole is larger for larger radius. Hence poles, in effect, "swamp" each other out. It appears that the radius should be chosen so that the pole does not extend its influence too far beyond the boundaries of its own region.

Convergence.

Only the boundaries, not the actual pole locations, are shown in figures 6 and 8, the plots for 200 passes. For R = 1, comparison of 5 and 6 indicate that the boundaries are converging nicely. Note table 1 indicates that 27 poles were stored in the first hundred passes, while only 12 were stored in the second hundreo. The boundaries in the region of interest are clearly better for 200 passes.

Not nearly as good is the case for R=4. There appears to be some convergence, since here 37 poles were added in the first hundred passes, while only 20 in the second hundred. However, the boundaries are not appreciably better. Furthermore, note that in figure 8, the computed △2 - △3 boundary has moved in toward the high probability region; many low probability points would be required to get it back out again.

Conclusions

The above results can be summarized as a tentative evaluation of the effects of these factors upon the algorithm.

Learning routine: not markedly important. Maximum routine perhaps slightly better.

Input Statistics: allowing for the fact that smaller σ results in a smaller region of interest, the algorithm seems to handle all cases equally as well.

Convergence: satisfactory, for proper choice of parameters.

Radius: extremely important. Should be chosen to be smaller than the "size" of the category set.

More Complicated Systems

Learning Routine

There seems to be no systematic relationship between W_0 and simulation accuracy. The C.G. scheme seems to have no clear advantage over the uniform scheme.

One reservation may be indicated. These results are for $N_{MAX} = 50$. The upward trend of the curves after saturation has been reached may indicate that the algorithm has done as well as it can for that form.

It is therefore concluded that there is no marked advantage of one over the other. Fewer calculations may be required for the maximum routine, while on the other hand, this routine requires a maximum selector, while the absolute routine may be realized with TLU's. It is assumed that in a specific problem, these and related considerations could be made the criterion of which to select.

Conclusions

In the study of the two-dimensional delay system, it was surmised that the radius factor R must be chosen to be smaller than the smallest "size" of a category set; and that choosing R larger than this size will degrade estimation, whereas choosing it smaller will not appreciably improve it. This seems to be supported by the data of the more complicated systems. We note also, that the large variance of the input, the less sensitive is the accuracy to R. Presumably this is because larger variance results in a probability of errors being corrected.

The steepness E is also a factor to be considered. Presumably, for R sufficiently small, E would not be an important parameter, whereas larger R may be partially compensated by large E., i.e., steeper decrease of $K(\bar{X}\ \bar{Y})$ with $\| \bar{x}-\bar{y} \|$.

The Effect of N_{MAX}

Figures 11 and 12 indicate the effects of increasing the maximum number of poles per category, N_{MAX}. Clearly, 30, 40 and 50 poles provide better accuracy than 20 poles, but 50 poles does not represent a marked improvement over N_{MAX} = 40 poles. Learning curves are presented in figure 11, and RMSE is plotted against N_{MAX} in 12.

It is assumed that increasing N_{MAX} will ultimately reduce the error; however, it is seen that the rate of reduction tends to decrease. The error due to averaging and quantizing, denoted by RMSAQ, is indicated on figure 11. This is a lower bound on identification error, since even if the identifier were to classify patterns correctly 100% of the time, the error would still be equal to RMSAQ.

Effect of Fine Quantization

Results

The second order system, test signal 6, was used to measure the effect of fine and coarse quantization on the algorithm's ability to simulate the system. For reference the minimum error due to averaging and quantization are repeated here from table 2.

Parameter values were:

ρ	=	0.5
σ	=	0.5
NMAX	=	50
NTAP	=	10
R	=	1.0
E	=	2.0

The absolute learning routine, CG weighting was employed.

The values of RMSAQ from table 2 for deck 6 were as follows:

NL	RMSAQ
5	.1049
15	.0474
25	.0339
35	.0319

The corrected RMS Error was arbitrarily defined as:

RMSE - RMSAQ,

that is, how close the actual RMSE came to the lower bound. Learning curves for 5, 15, 25, and 35 levels, plotting corrected RMS error is shown in figure 13. Note that the corrected RMSE increases rather than decreases for increasing NL. Uncorrected data are not presented, but can easily be inferred from figure 13 and the above table. The face is, that in all cases, the uncorrected curves were close together.

This result is not as surprising as it might first appear. Figure 14 gives data on the ratio of correct classifications for the several values of NL. That is, the ratio of correct estimate is that fraction of patterns presented by the test deck that were correctly classified. Clearly, the fewer levels there are, the more reliable will be the estimation.

One reservation must be noted. As the number of levels is increased, the extent of each category set is decreased. Hence, the radius factor R should be decreased accordingly. The curves of figures 13 and 14 were all for R = 1, which is probably far from optimum for 25 and 35 levels. More studies of this point would probably be helpful.

Conclusions

The main conclusion is that it is by no means clear that identification error can be reduced by simply increasing the fineness of quantization. This is probably true in general for the method of potential functions, as well as for most other modal learning techniques. Furthermore, a similar mechanism may be present in other types of learning algorithms as well.

Probably, the complication introduced by finer quantization can be mitigated by optimizing the Radius and Steepness factors; however, it is not clear that this will provide the whole answer.

Conclusions

The Experimental Results

The results indicate that the most important parameters of the algorithm are the radius and steepness factors R and E, the number of quantization levles NL, and the maximum number of poles per category N_{MAX}. The last two are important also in that they determine the amount of storage required for realizing the algorithm. These 4 factors are furthermore closely related. For example, increasing N_L probably requires a change in R and E. It is noted that increasing N_L is probably a less effective means of improving accuracy than is increasing N_{MAX}.

The results indicate that a choice of the absolute or maximum learning routine is not important to the accuracy or the storage requirements, nor is choice of a pole weighting scheme. Hence, these may be chosen to satisfy other criteria, such as hardware simplicity.

Choice of Tap interval, and settling time are important, but are properties of the system, rather than of the algorithm.

Designing a Potential Function Simulator for a Given System.

In light of the above, the important steps in designing an adaptive simulator employing the potential function algorithm are as follows:

1. N_L is chosen as the smallest number such that the quantization error is satisfactorily small. Discussions of Max (9) and of Widrow (13), (14) might provide a criterion.

2. The number of taps and the settling time is chosen according to what knowledge about the input is available.

3. R and E are chosen. R should be small enough as discussed above, and in general is a function of N_L. A first approximation

might be as follows:

The "volume" of a hypersphere in d-dimensional space is given by (7).

$$V = \frac{2\ R^{d}}{d}\ \frac{{}^{d/2}}{(1/2d)}$$

Hence, if it is known that the volume of sample space, as determined by the variance of the input is V', then R is chosen such that

$$V = \frac{1}{N_L}\ V',$$

in other words, to a first approximation, it is assumed that each category subset occupies an equal volume in pattern space.

4. N_{MAX} is chosen to be the largest number for feasible memory requirements.

5. The other parameters are chosen to suit whatever criteria are appropriate.

Suggestions for Further Research

Two topics might be appropriate for further study. The relationship between R and N_L might be investigated, and the effect upon identification of increasing N_L while optimizing R. Also investigations of pole weighting for smaller values of N_{MAX} may be of interest.

Finally, it is noted that the algorithm described here is the basic algorithm, presented in the first paper of Aizerman *et al*. The two subsequent papers discuss several variations, including a probablistic technique which should be highly appropriate for adaptive simulation.

BIBLIOGRAPHY

1. Aizerman, M. A., Braverman, E. M. and Rozoner, L. I. "Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning". Automation and Remote Control (Avtomatika i Telemekhanika) vol. 25, no. 6, June 1964, pp. 821-837 (English).

2. ____________, "The Probability Problem of Pattern Recognition Learning and the Method of Potential Functions", Automation and Remote Control (Avtomatika i Telemekhanika), vol. 25, no. 9, Sept. 1964, pp. 1307-1327 (English).

3. ____________, "The Method of Potential Functions for the Problem of Restoring the Characteristic of a Function Converter from Randomly Observed Points", Automation and Remote Control (Avtomatika i Telemekhanika) vol. 25, no. 12, December 1964, pp. 1546-1556 (English).

4. Bashkirov, O. A., Braverman, E. M., and Muchnik, I. B., "Potential Function Algorithms for Pattern Recognition Learning Machines", Automation and Remote Control (Avtomatika i Telemekhanika), vol. 25, no. 5, May 1964, pp. 629-631 (English).

5. Braverman, E. M., "Experiments on Machine Learning to Recognize Visual Patterns", Automation and Remote Control (Avtomatika i Telemekhanika), vol. 23, no. 3, March 1962, pp. 315-327 (English).

6. Cooper, P. W., "Hyperplanes, Hyperspheres and Hyperquadric as Decision Boundaries", in Tou and Wilcox, (Eds.), Computer and Information Sciences, Spartan Books, Washington, D. C., 1964.

7. Kendall, M. G., A Course in the Geometry of n Dimensions, Hafner Publishing Company, New York, 1961.

8. Lewin, I. Y. and Sapozhnikov, L. B., "Recognition Algorithms", Automation and Remote Control (Avtomatika i Telemekhanika), vol. 24, no. 6, June 1963, pp. 769-773 (Engiish).

9. Max, J., "Quantizing for Minimum Distortion", IRE Trans. Info. Theo., March 1960.

10. Nilsson, N. J., Learning Machines, McGraw-Hill, New York, 1965.

11. Sebestyan, G. S., Decision Making Processes in Pattern Recognition, MacMillan Company, New York, 1962.

12. Vapnik, V. N. and Lerner, A. Ya, "Pattern Recognition Using Generalized Portraits", Automation and Remote Control (Avtomatika i Telemekhanika), vol. 24, no. 6, June 1963, pp. 709-715, (English).

13. Widrow, B. "A Study of Rough Amplitude Quantization by Means of Nyquist Sampling Theory", Trans. IRE, vol. CT 3, no. 4., December 1956.

14. Widrow, B., "Statistical Analysis of Amplitude Quantized Sampled-Data Systems", AIEE Trans. paper no. 60-1240.

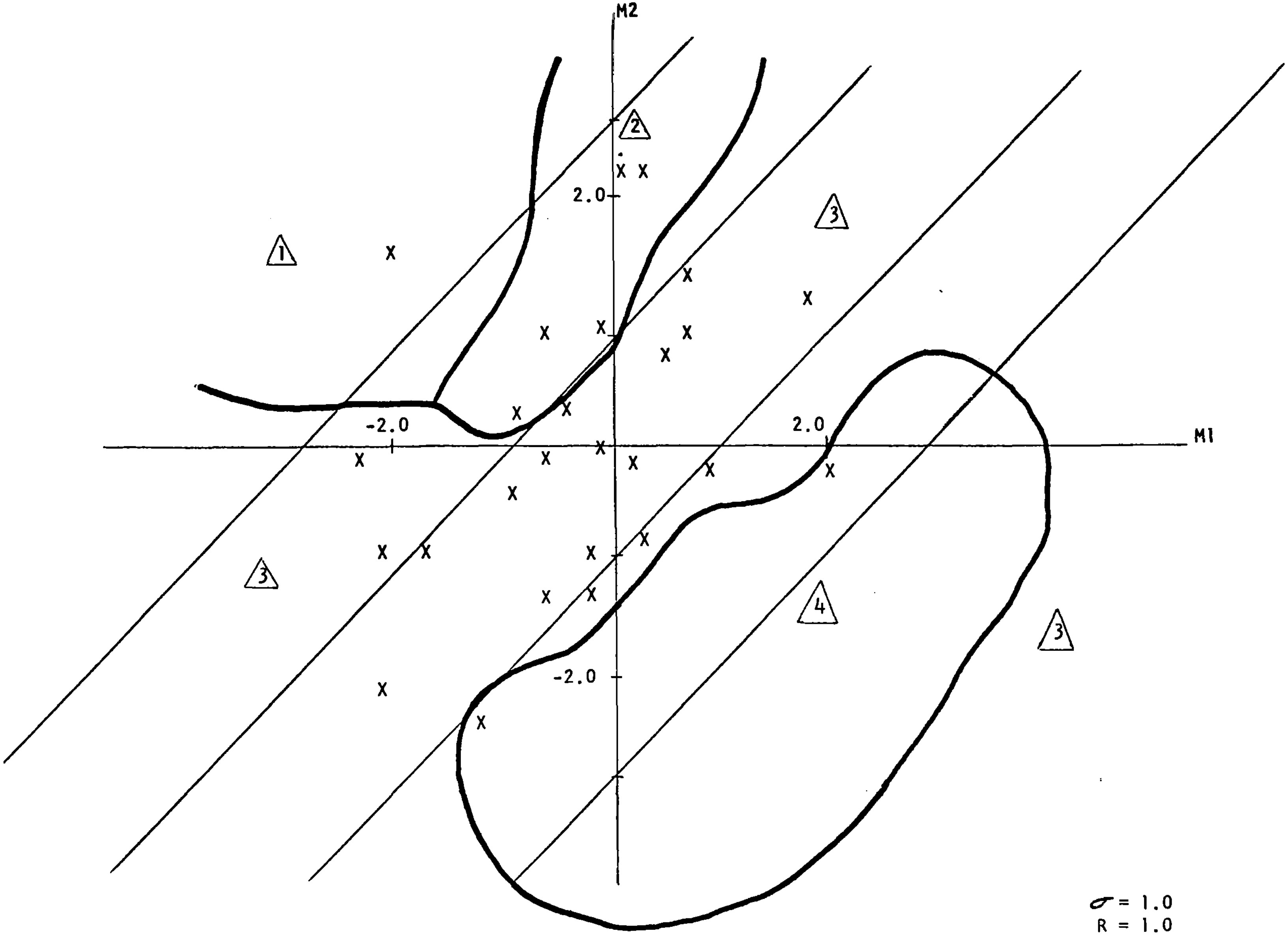


FIGURE 5 - Two-Dimensional Delay System

σ = 1.0
R = 1.0
100 Patterns
Maximum Routine

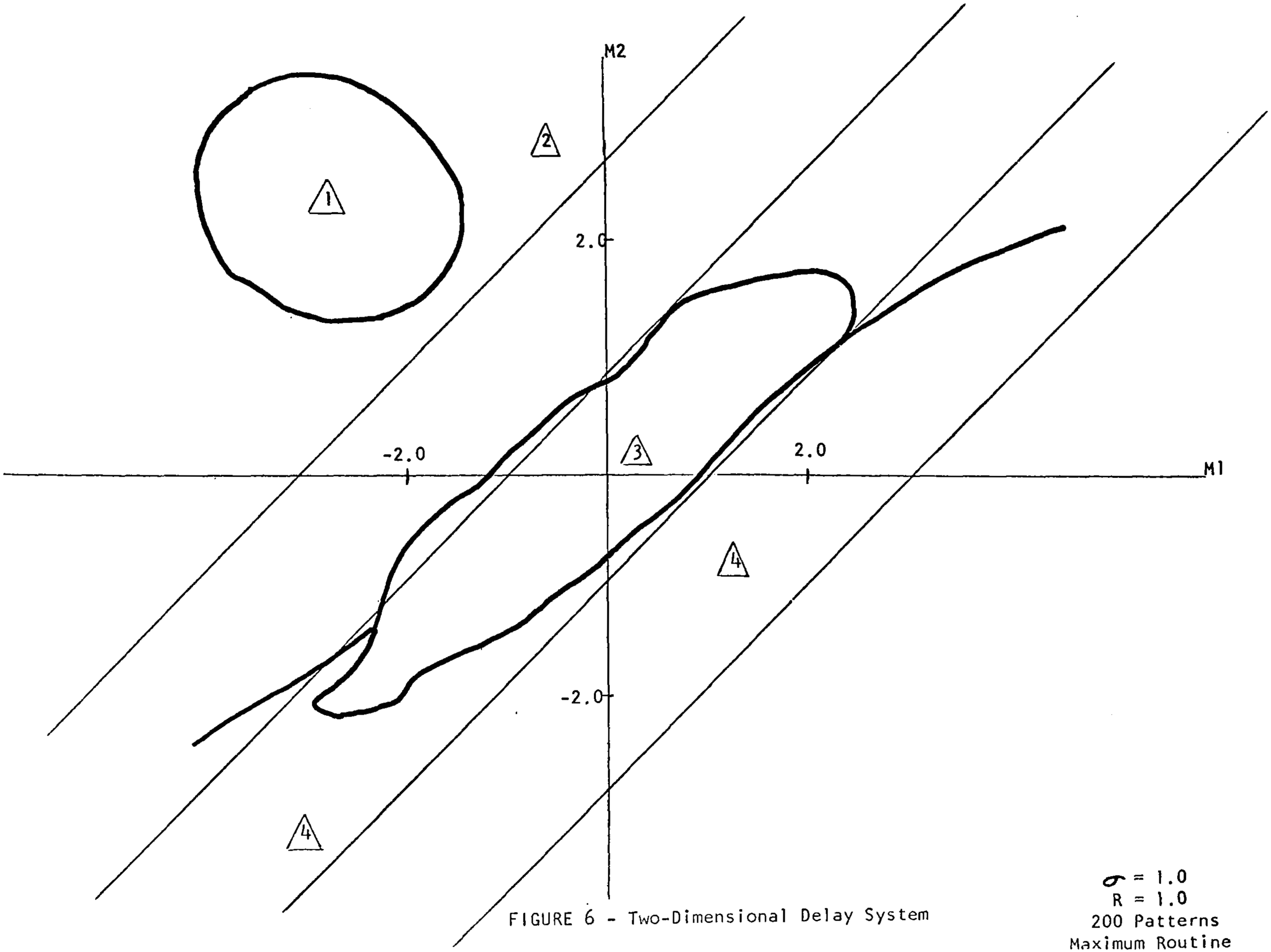


FIGURE 6 - Two-Dimensional Delay System

σ = 1.0
R = 1.0
200 Patterns
Maximum Routine

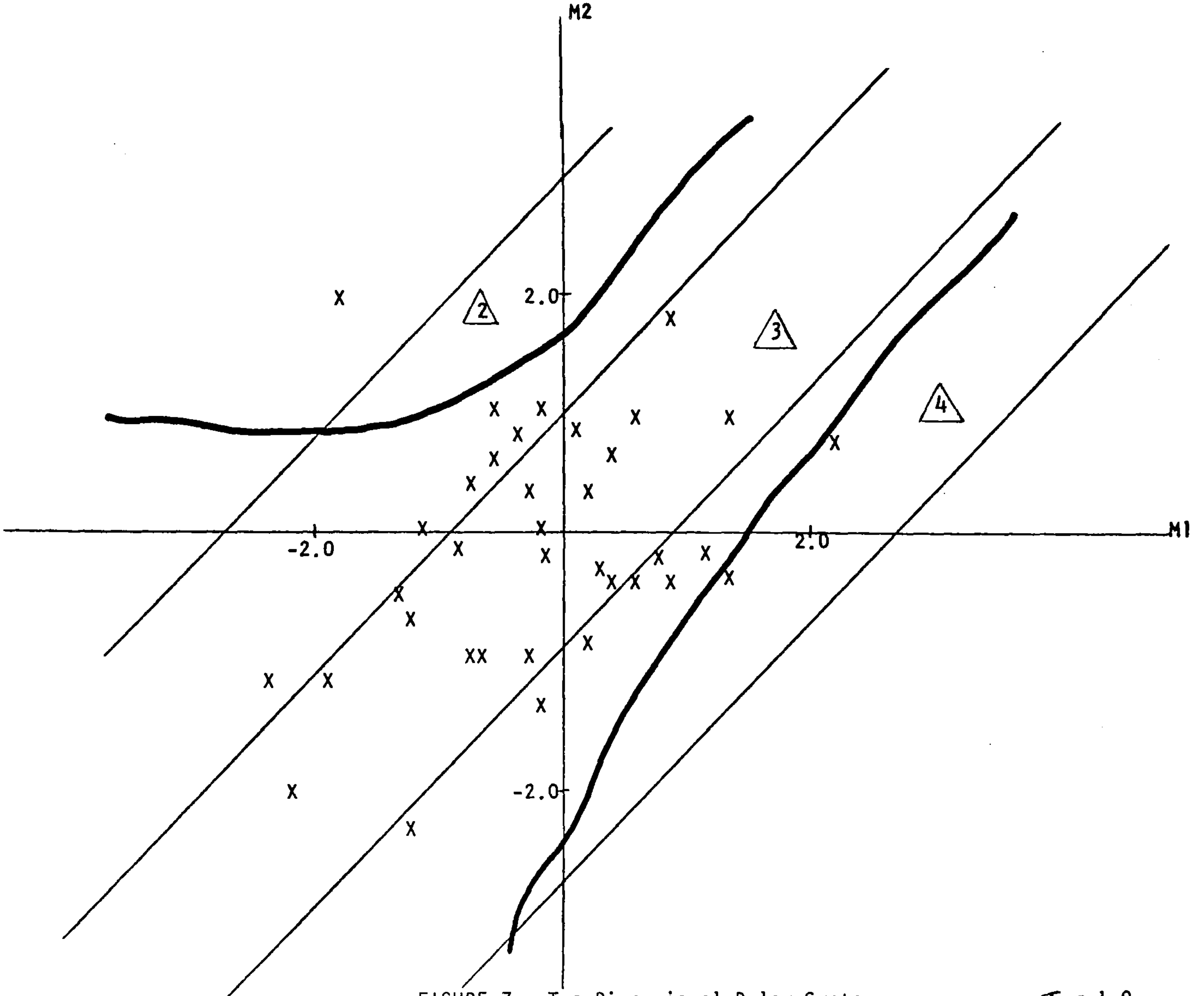


FIGURE 7 - Two-Dimensional Delay System

$\sigma = 1.0$
$R = 4.0$
100 Patterns
Maximum Learning

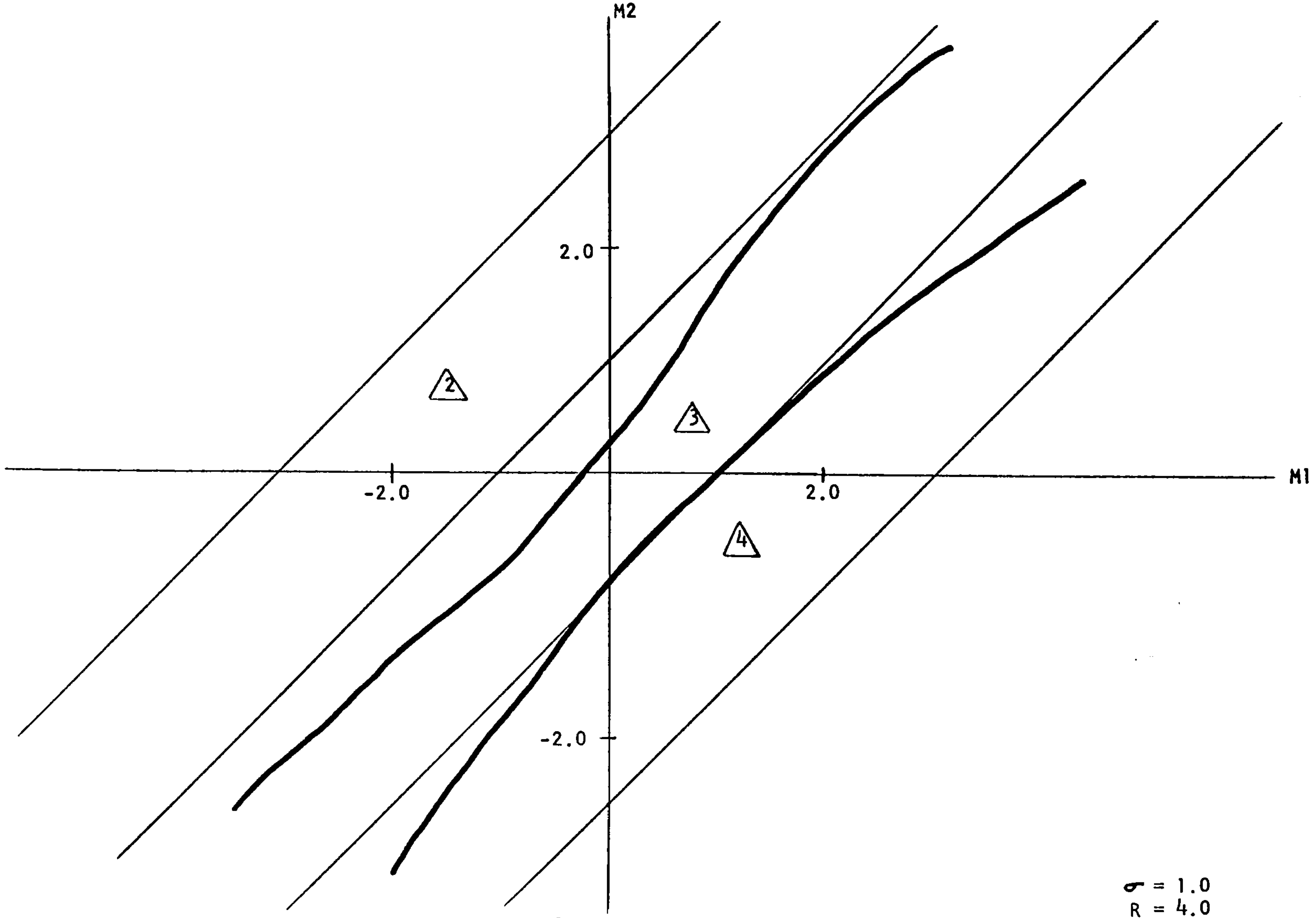


FIGURE 8 - Two-Dimensional Delay System

σ = 1.0
R = 4.0
200 Patterns
Maximum Learning

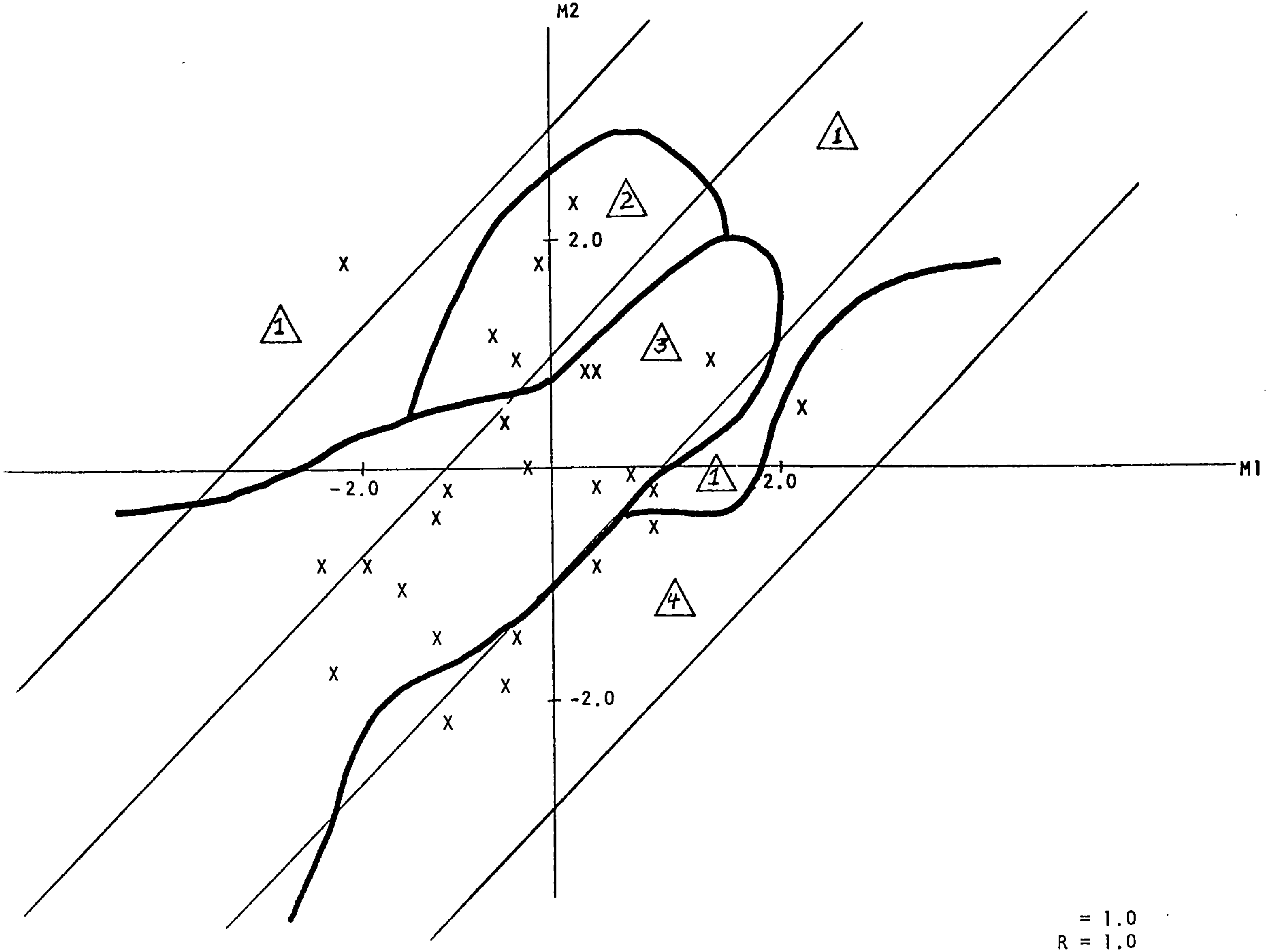
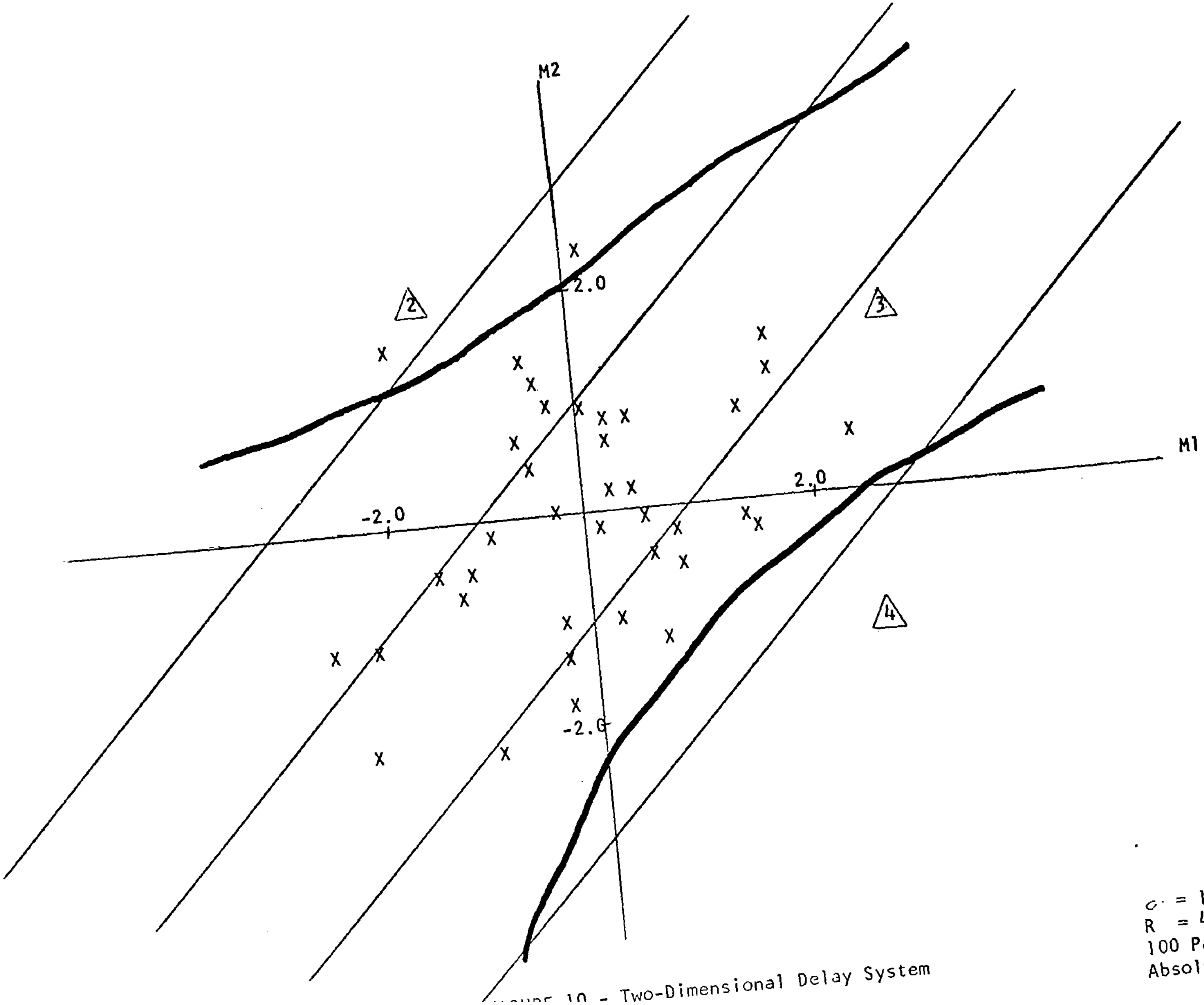


FIGURE 9 - Two-Dimensional Delay System

= 1.0
R = 1.0
100 Patterns
Absolute Routine



IGURE 10 - Two-Dimensional Delay System

σ = 1.0
R = 4.0
100 Patterns
Absolute Learning

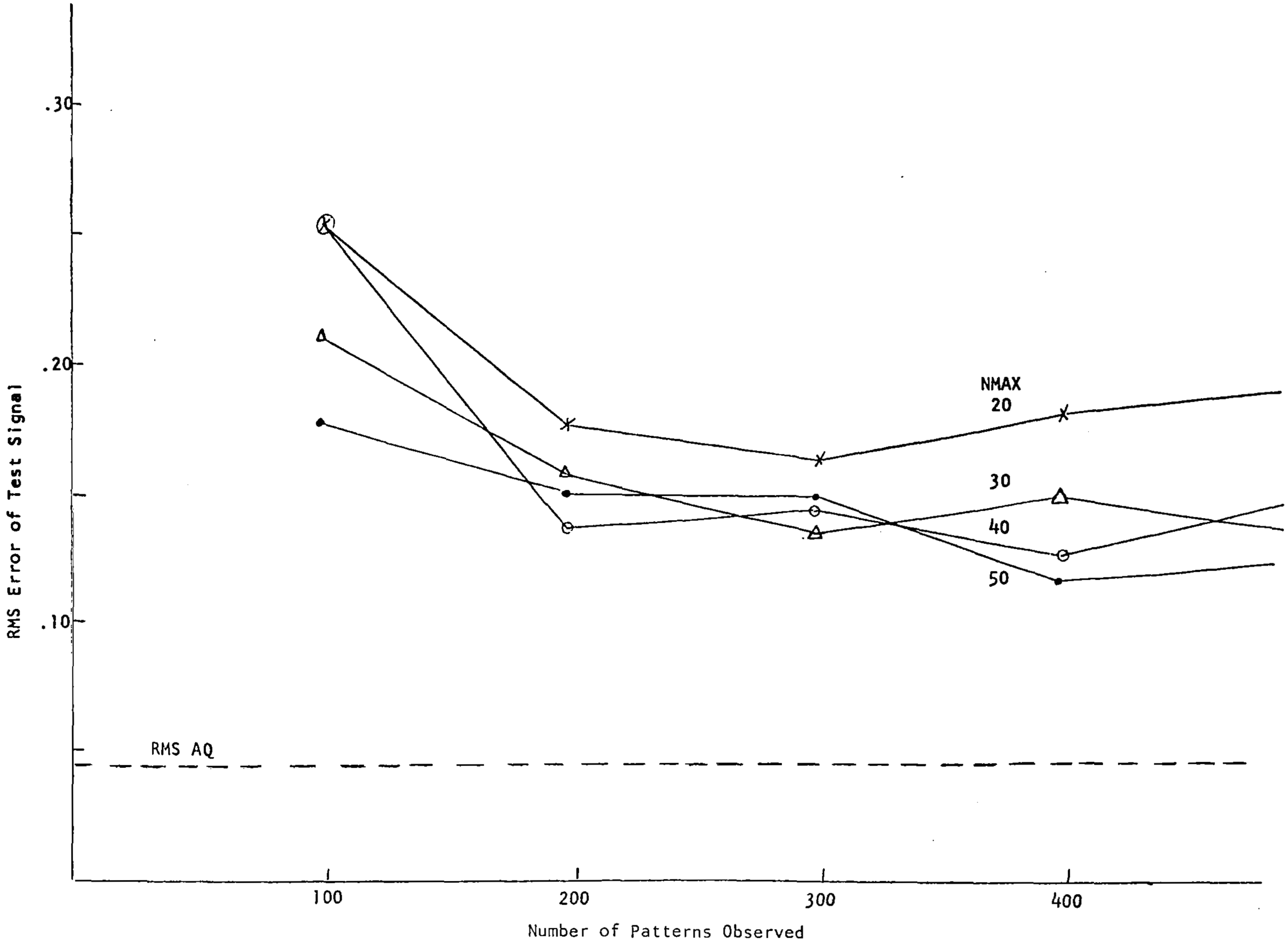


FIGURE 11 - Learning Curve
Second Order

15 levels
P = 0.5
σ = 0.5

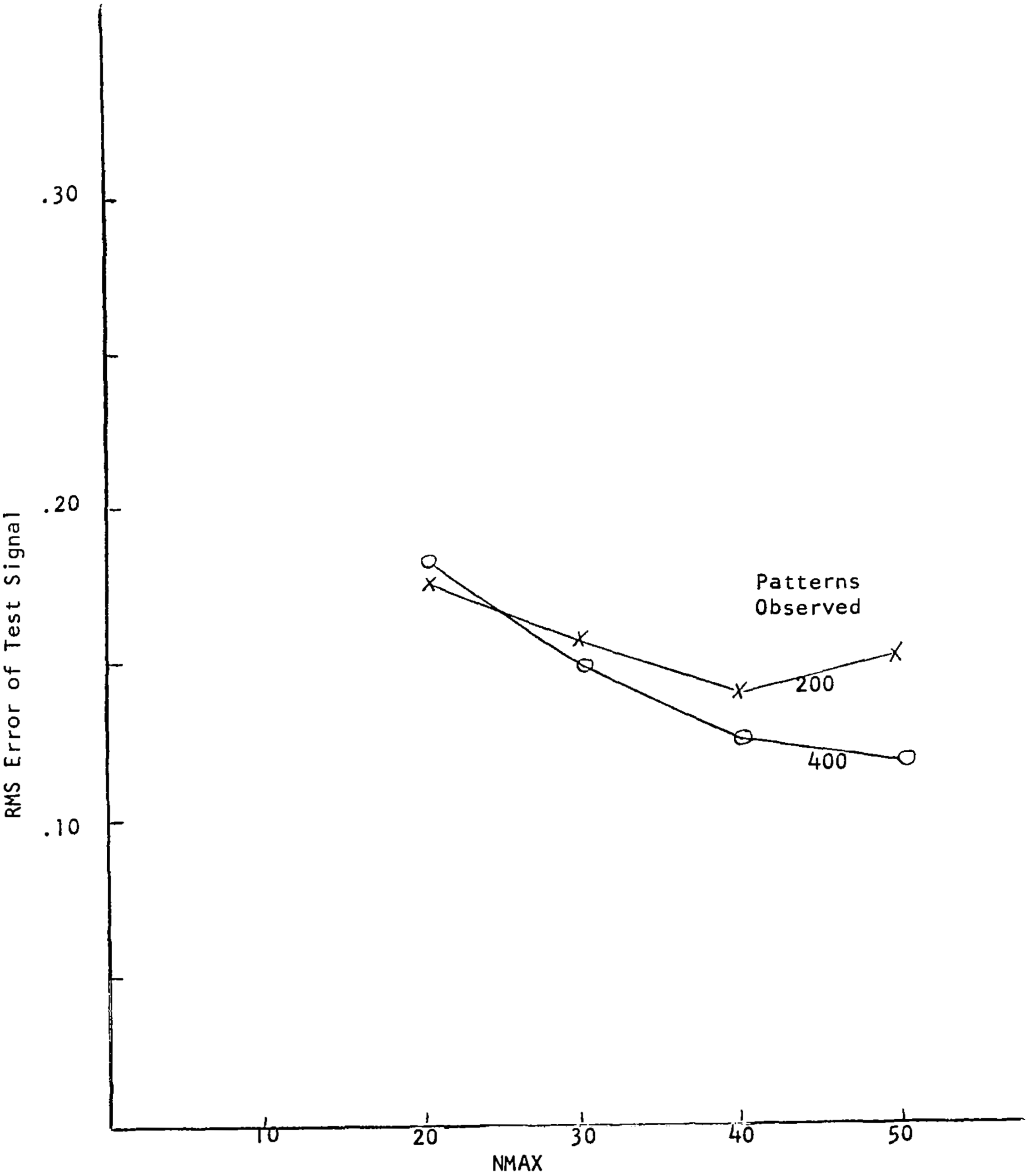
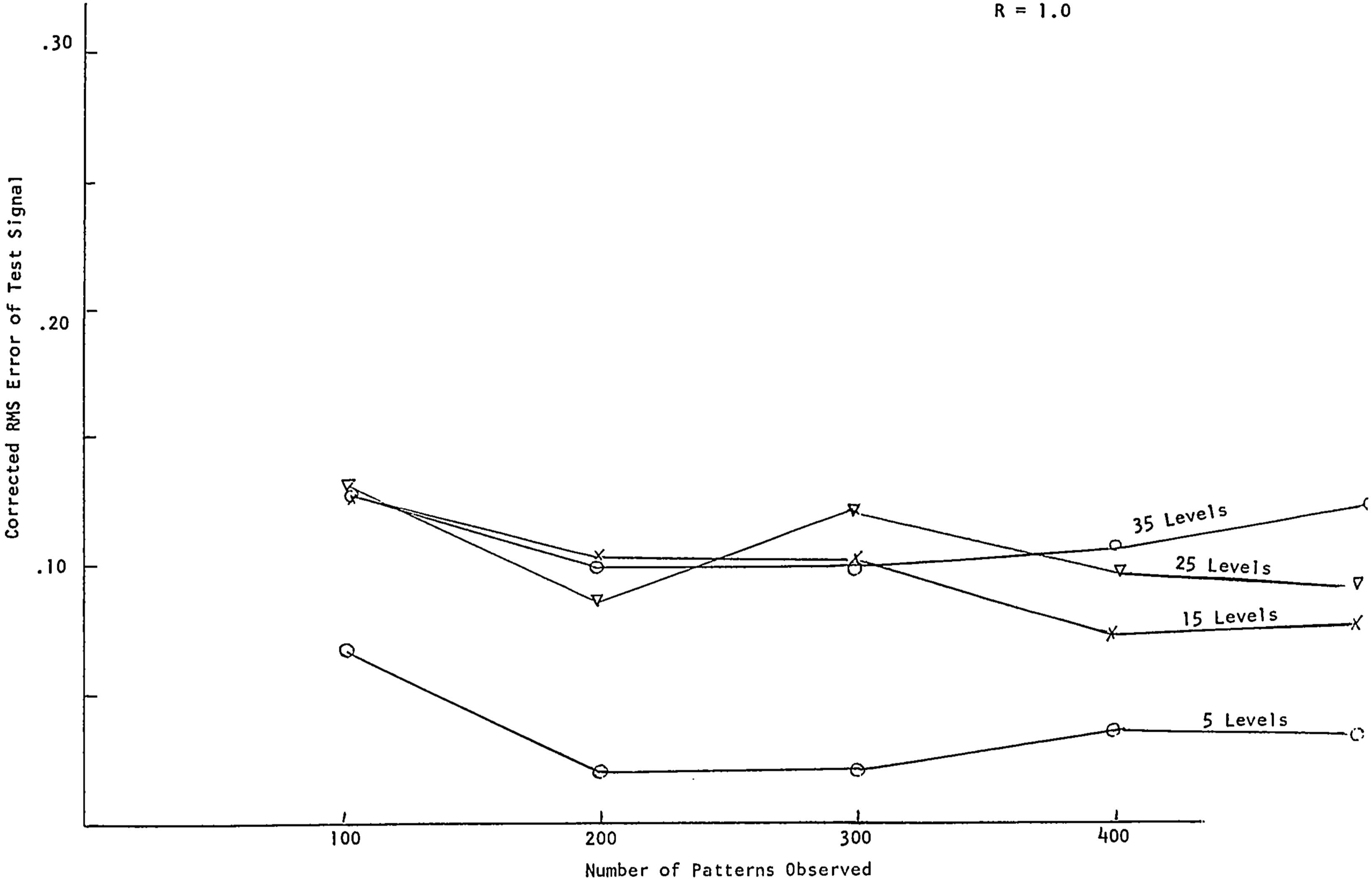


FIGURE 12 - Learning Error vs. Maximum Number of Poles per Category
Second Order System
$\rho = 0.5$
$\sigma = 0.5$
$R = 1.0$

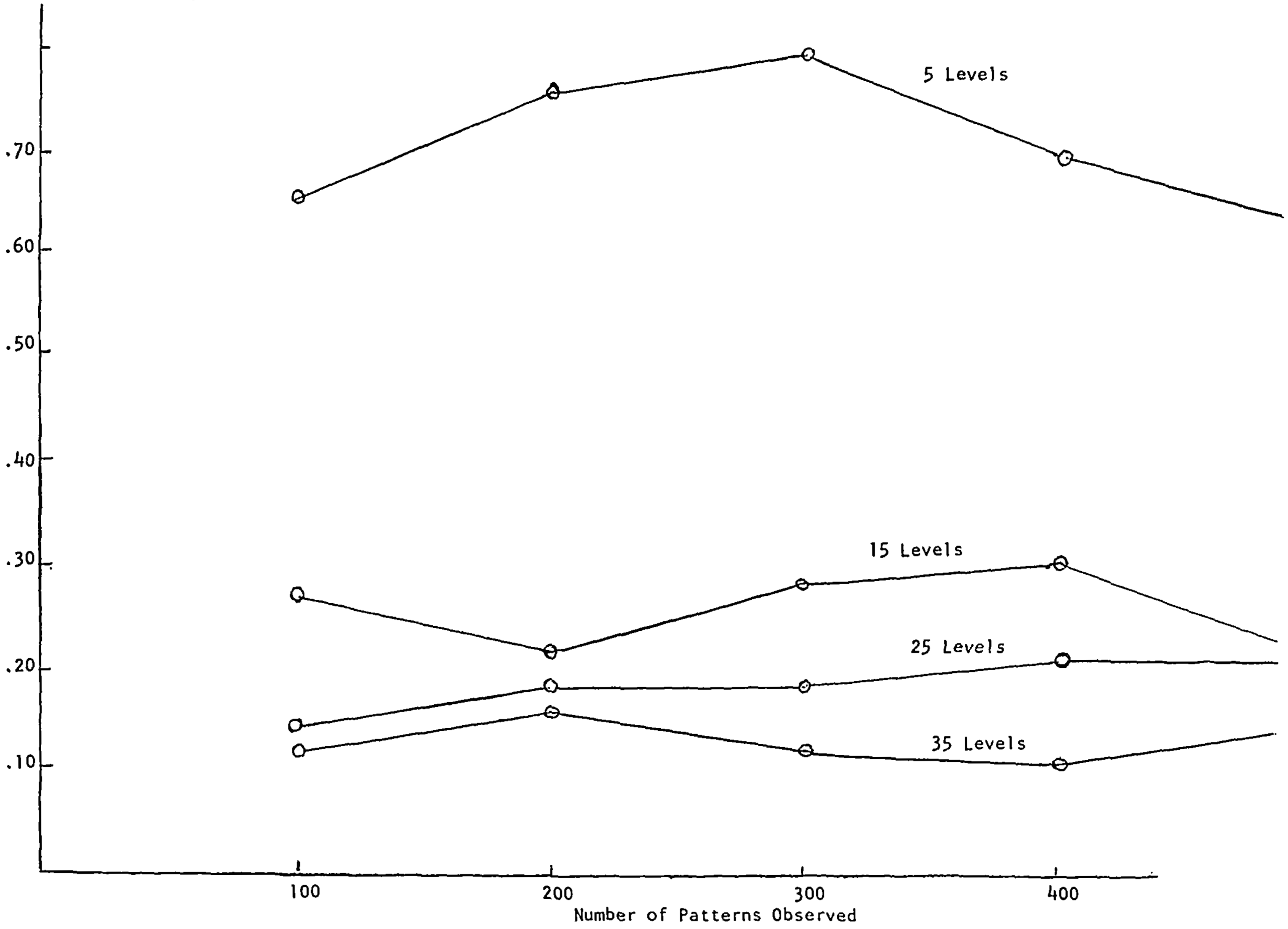
FIGURE 13 - Learning Curve Second Order System

Corrected RMS Error
vs. Number of Patterns
Observed for Several Degrees
of Quantization
P = 0.5
σ = 0.5
R = 1.0



Ratio of Correct Estimates for
Several Degrees of Quantization
$\sigma = 0.5$
$P = 0.5$
$R = 1.0$

FIGURE 14 - Learning Curve
Second Order System


5 Levels
15 Levels
25 Levels
35 Levels
Number of Patterns Observed